

**Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки**

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Проектування інформаційно-управляючої системи мінімаркету»

Студента 2м курсу, 23 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Тищенко Дмитра
Олександровича

Науковий керівник
доктор технічних наук,
професор,
завідувач кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Криворучко Олена
Володимирівна

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис гаранта

Токар Володимир
Володимирович

КИЇВ – 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«29» грудня 2020 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Завдання

на випускний кваліфікаційний проєкт студентіві

Тищенко Дмитра Олександровича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Проектування інформаційно-управляючої системи мінімаркету»

Затверджена наказом ректора від «28» грудня 2020 р. № 3923

2. Строк здачі студентом закінченого проєкту 25 листопада 2021р.

3. Цільова установка та вихідні дані до проєкту

Мета дослідження є проектування інформаційно-управляючої системи мінімаркетом для оптимізації та спрощення продажів. Основним завданням є забезпечення максимально швидкої обслуги клієнта, продажі великої кількості товару максимально зручним для мінімаркету та клієнта способом.

Об'єкт дослідження: процеси проективання інформаційно-управляючої системи.

Предмет дослідження: інформаційно-управляюча система мінімаркеті.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ВИМОГИ ДО ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МІНІМАРКЕТУ

1.1. Аналіз предметної області та постановка задачі

1.2. Огляд вимог до програмного забезпечення

1.3. Розгляд аналогів ПЗ

1.4. Вибір моделі розробки ІУС

1.5. Середовище розробки Visual Studio

1.6. Висновок до розділу 1

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЮЧОЇ СИСТЕМИ

2.1. Виявлення ризиків при проектуванні ІУС

2.2. Пошук акторів системи»

2.3. Виявлення основних варіантів використання

2.4. Проектування діаграми послідовності

2.5. UML-діаграми активності

2.5. Проектування бази даних

2.6. Висновок до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МІНІМАРКЕТУ

3.1. Розробка та тестування головних функцій системи

3.2. Тестування системи та інших функцій

3.3. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра</i>	29.12.2020	29.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. Вимоги до функціонування інформаційно-управляючої системи мінімаркету</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. Проектування інформаційно-управляючої системи</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. Реалізація та тестування інформаційно-управляючої системи мінімаркету</i>	21.06.2021	21.06.2021
9.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
10.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
11.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
12.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
13.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
14.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
15.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021 – 25.11.2021	22.11.2021
16.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
17.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
18.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «29» грудня 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту Криворучко О.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Тищенко Д.О.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту

(підпис, дата)

Відмітка про попередній захист Криворучко О.В.

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента Тищенко Д.О.

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

Завідувач кафедри Криворучко О. В.

(підпис, прізвище, ініціали)

« » 20 р.

АНОТАЦІЯ

Випускний кваліфікаційний кваліфікаційний проєктна здобуття освітнього ступеня магістр за спеціальністю 121 – «Інженерія програмного забезпечення». – Київський національний торговельно-економічний університет, Київ, 2021 р.

Метою випускного кваліфікаційного кваліфікаційного проєкту є проєктування та впровадження інформаційно-управляючої системи мінімаркета для оптимізації роботи з продажами на базі програмного середовища C# .Net. Система повинна забезпечити максимально зручний та швидкий спосіб продажі товарів та ведення усієї звітності пов'язаною з цією діяльністю.

Під час виконання даної роботи застосовано мову програмування C# та технології .Net. Середовище розробки Visual Studio дозволили використати усі переваги розробки графічного дизайну з допомогою Windows Form. Для зберігання даних було обрано базу даних MySql. При виконанні даної роботи розглянуто існуючі програмні комплекси та проведено їх аналіз. Проаналізовано всі їх переваги та недоліки. Результатом розробки стало програмне забезпечення яке дозволяє спростити продаж товарів, та дозволяє максимально швидше обслуговувати клієнта мінімаркету. Проведено проєктування баз даних та інформаційно-управляючої системи.

Ключові слова: WINDOWS, MYSQL, WINDOWS FORM, C# .NET, бази даних, інформаційно-управляюча система, інструмент автоматизації.

ABSTRACT

Graduation qualification project qualification master's degree in specialty 121 - "Software Engineering". - Kyiv National University of Trade and Economics, Kyiv, 2021

The purpose of the final qualification project is to design and implement an information and management system of the mini-market to optimize sales with a C # .Net software environment. The system should provide the most convenient and fast way to sell goods and keep all records related to this activity.

C # programming language and .Net technology were used in this work. The Visual Studio development environment allows you to take full advantage of graphic design development using Windows Form. The MySql database was selected for data storage. In performing this work, the existing software packages are considered and analyzed. All their advantages and disadvantages are analyzed. The result of the development was software that simplifies the sale of goods, and allows you to serve the customer of the mini-market as quickly as possible. The design of databases and information and management system.

Keywords: WINDOWS, MYSQL, WINDOWS FORM, C # .NET, databases, information management system, automation tool.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІУС – інформаційно-управляюча система

UI (User Interface) – користувацький інтерфейс

ПЗ – програмне забезпечення

БД – база даних

SQL – (structured query language)

CLI – (Common Language Infrastructure)

API – (Application Programming Interface)

CLI – це платформа, на якій виконуються програми .Net.

					<i>КНТЕУ 121 02з-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		22.12.20	Проектування інформаційно- управляючої системи мінімаркету	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		22.12.20		ПС	2	63
Гарант		Токар В.В.		22.12.20		Факультет інформаційних технологій 2з курс, 2м група		
Розробив		Тищенко Д.О.		22.12.20				
					<i>Перелік умовних позначень, символів, одиниць, скорочень і термінів</i>			

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 ВИМОГИ ДО ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МІНІМАРКЕТУ	7
1.1. Аналіз предметної області та постановка задачі	7
1.2. Огляд вимог до програмного забезпечення	9
1.3. Розгляд аналогів ПЗ	10
1.4. Вибір моделі розробки ІУС	14
1.5. Вибір технологій та інструментів для розробки	17
1.6. Середовище розробки Visual Studio	24
1.7. Висновки до Розділу 1	26
РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ	27
2.1 Виявлення ризиків при проектуванні ІУС	27
2.2. Пошук акторів системи	28
2.3. Виявлення основних варіантів використання	29
2.4. Проектування діаграми послідовності	41
2.5. UML-діаграми активності	43
2.6. Проектування бази даних	46
2.7. Висновки до Розділу 2	52
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МІНІМАРКЕТУ	53
3.1. Розробка та тестування головних функцій системи	53
3.2. Тестування системи та інших функцій	60
3.3. Висновки до Розділу 3	59
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
ТЕХНІЧНЕ ЗАВДАННЯ	69

					<i>КНТЕУ 121 023-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		27.02.21	Проектування інформаційно-управляючої системи мінімаркету	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		27.02.21		Зміст	3	63
Гарант		Токар В.В.		27.02.21		Факультет інформаційних технологій 23 курс, 2м група		
Розробив		Тищенко Д.О.		27.02.21				
					Зміст			

ВСТУП

Експлуатація мінімаркету вимагає отримання багатьох адміністративних, управлінських та маркетингових навичок. Зважаючи на те, що достатня кількість записів для складання щомісячних звітів про продажі повинна бути опрацьованою та систематизована, ці навички необхідні для забезпечення безперебійності роботи мінімаркету. Окрім цього варто враховувати велику кількість товару то його систематизацію. У вирішенні цих проблем може допомогти автоматизована система управління продажами мінімаркету.

Інформаціо-управляюча система (ІУС) - це місце, де ваш клієнт здійснює оплату за товари чи послуги у вашому магазині, а також місце касира з касовим апаратом який допоможе клієнтові здійснити оплату.

Наскільки очевидними є переваги ІУС, статистика показує, що 56 відсотків роздрібних продавців все ще не використовують її. Натомість, багато хто все ще використовує комбінацію ручних методів, касових апаратів, QuickBooks та Excel для ведення бухгалтерського обліку.

То чому продавці ще не зробили цього кроку до ІУС? Для початку впровадження нової технології - особливо технології, яка є центральною у вашому бізнес-процесі - може бути страшною та непосильною. Роздрібна торгівля повинна враховувати негативні наслідки відсутності ІУС.

Гнучкість у роботі є ключовою. Варто завжди переконуватися, що постачальник системи працює з платіжним процесором / шлюзом на вибір, щоб була можливість контролювати витрати. Також варто переконатися, що ІУС може безперешкодно інтегруватися, щоб не було перебоїв у роботі мінімаркету.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 023-1.МР</i>			
Зав. каф.		Криворучко О.В.		27.02.21	<i>Проектування інформаційно-управляючої системи мінімаркету</i>	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		27.02.21		Вступ	4	63
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій 23 курс, 2м група</i>		
Розробив		Тищенко Д.О.		27.02.21				
					<i>Вступ</i>			

Варто також врахувати, що впровадження системи управління в мінімаркеті вимагає певного устаткування та умов.

На сьогоднішній день впровадження таких систем стає все більше актуальним, адже асортимент товарів та послуг невпинно розширюється, а тримати у пам'яті усі ціни просто неможливо.

Метою ВКП є проектування інформаційно-управляючої системи мінімаркетом для оптимізації та спрощення продажів. Основним завданням є забезпечення максимально швидкої обслуги клієнта, продажі великої кількості товару максимально зручним для мінімаркету та клієнта способом.

Для того, щоб досягти вище вказану мету необхідно вирішити такі завдання:

- дослідити предметну область та здійснити постановку проблеми;
- побудувати модель автоматизованої системи;
- реалізувати програмне забезпечення для автоматичної системи управління;
- провести тестування розробленої системи.

Проаналізувавши предметну область, можемо виділити перелік задач, які необхідно вирішити:

- Розробка архітектури системи;
- Проектування бази даних;
- Реалізація роботи сканера штрих кодів;
- Реалізація системи;
- Тестування системи.

Дана тема є актуальною, оскільки на даний момент існує велика кількість мінімаркетів та магазинів, які потребують оновлення або покращення програмного забезпечення. Велика кількість інформації про товари зберігається в незручному та важкодоступному вигляді, а користувацький інтерфейс ускладнює навчання користувачів даними системами. Важливим також є ведення історії

					КНТЕУ 121 02м-16.МР	Аркуш
						5
Зм.	Аркуш	№ докум	Підпис	Дата		

продажів для отримання звітності про прибутковість магазину чи мінімаркету. В даному програмному забезпеченні також реалізовано можливість додавання нового касира в систему, а також отримання інформації та звітності про продажі.

З введенням подібної системи у роботу буде вирішено вище згадані проблеми, а також швидке і інтуїтивно зрозуміле керування автоматичною системою управління.

Для написання ВКП було обрано мову програмування C#. Вибір було зроблено у зв'язку з зручними інструментами, які нам надає ця мова для написання графічного інтерфейсу та роботи з базою даних. Як середовище розробки було вибрано Visual Studio 2017 та технологію Windows Forms.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						6
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 1

ВИМОГИ ДО ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МІНІМАРКЕТУ

1.1. Аналіз предметної області та постановка задачі

Завданням магістерської роботи є розробка автоматичної системи управління мінімаркетом для оптимізації та покращення продажів. Система повинна мати зручний та зрозумілий інтерфейс який забезпечить продуктивну роботу користувача та спростить продаж товарів.

Програмні засоби системи повинні забезпечувати справне її функціонування в нормальних умовах. Також потрібно передбачити можливість того, що буде одночасне звернення багатьох запитів до неї. Система повинна гарантувати безпеку введених даних та виконувати те, що від неї вимагають користувачі. Ключовою є можливість роботи з сканером штрих кодів та бар кодів, оскільки саме вони забезпечують максимально швидку та комфортну роботу з системою [16]. До основних функцій системи можна віднести:

- ✓ Робота з сканером штрих кодів та бар кодів;
- ✓ Швидкий пошук товару за кодом;
- ✓ Збільшення асортименту товарів в системі;
- ✓ Ведення звітності продажів;
- ✓ Можливість автоматично вирахувати знижку за допомогою дисконтної карти постійного клієнта;
- ✓ Підтримка роботи декількох користувачів. Дана предметна область є дуже актуальною, оскільки ключовим завданням мінімаркету вважається вирішення та задоволення базових і не тільки потреб покупців.

					<i>КНТЕУ 121 023-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		16.04.21	<i>Проектування інформаційно-управляючої системи мінімаркету</i>	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		16.04.21		РІ	7	63
Гарант		Токар В.В.		16.04.21		<i>Факультет інформаційних технологій 23 курс, 2м група</i>		
Розробив		Тищенко Д.О.		16.04.21				
					<i>Вимоги до функціонування інформаційно-управляючої системи мінімаркету</i>			

Великий асортимент базових товарів, мінімаркет може з легкістю замінити інші, менші торгові точки та магазини із спец товарами. Забезпечуючи великий обсяг продукції, зазвичай мінімаркети витісняють привичні для більшості людей магазини, тому даний напрямок розвивається і виникає гостра потреба в автоматизації та розробці додаткового ПО [14].

Сутність постановки задачі полягає в розробці програмного забезпечення для ПК, яке повинне спростити та пришвидшити продаж товарів в мінімаркеті. На рисунку 1.1. зображено схему роботи мінімаркету.



Рис. 1.1. Схема роботи мінімаркету

Сучасний мінімаркет це великий магазин з найрізноманітнішим асортиментом товарів. Для швидкого продажу та оновлення даних про ці товари виникає необхідність в розробці сучасних програмних систем. Тому виникають певні вимоги до програмних систем. Під час створення даної програми потрібно забезпечити максимальну швидкість взаємодії з користувачем, а також простоту і зручність використання користувацького інтерфейсу [17].

					КНТЕУ 121 02з-16.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

1.2. Огляд вимог до програмного забезпечення

ГУС мінімаркетом – це система яка спрощує та прискорює обслуговування клієнтів. Саме тому то цієї системи висувається ряд вимог, які повинні забезпечувати якісне та коректне функціонування програми. Цей тип вимог описує, що система повинна робити. В переліку нижче перераховано ключові функціональні вимоги до системи:

- кожна функція повинна бути коректно реалізована та виконувати те, що від неї вимагається;
- вибір товару за допомогою штрих коду;
- робота системи з базами даних MySQL;
- можливість додати в чек одночасно більше 30 товарів;
- час на додавання одиниці товару не повинен перевищувати 0,5 секунди;
- сканування дисконтної карти за допомогою сканера штрих кодів або при допомозі вводу ідентифікаційного коду картки;
- додавання декількох користувачів в систему;
- формування звітності по продажах за певний період;
- пошук товару за допомогою коду;
- система автоматичну вираховує решту, яку потрібно видати клієнту.

Нефункціональні вимоги

Окрім функціональних вимог, які було описано вище, також велику роль відіграються нефункціональні вимоги. На відміну від описаного вище, нефункціональні вимоги описують не те, що повинна робити система, а те як вона повинна це робити [16]. Ключовими вимогами до системи є лаконічний та зрозумілий користувацький інтерфейс.

Ключові дані списку товарів повинні бути представлені у вигляді таблиці.

					КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

1.3. Розгляд аналогів ПЗ

Програма ShopDesk

ShopDesk – один з аналогів АСУ мінімаркету. Офісна система, для спрощення та вдосконалення роботи касир в магазинах з різною специфікою, а також з самостійним обслуговуванням. Дозволяє інтегруватися в систему торгового центру, представлення єдиним модулем, який дозволяє працювати з різноманітним обладнанням, в тому числі торговими.

Система не є вимогливою до швидкісного каналу з базою даних (при роботі з MSSQL Server), ця особливість дозволяє без проблем налагодити віддалений доступ через інтернет. Окрім цього, також реалізовано функцію роботи з базою даних навіть при відсутності безперервного з'єднання з базою даних на сервері через інтернет. Реалізована підтримка з різними реєстраторами, а також з фіскальними, а також можливість роботи з картками знижок для постійних клієнтів, надання звітів різного типу. Можливість широкою кастомізації роботи касира та ведення повної звітності його дій

Робота додатку відбувається з базою даних безпосередньо, завдяки чому стає можливим отримувати найактуальнішу та найбільш достовірну інформацію про наявний в магазині товар.

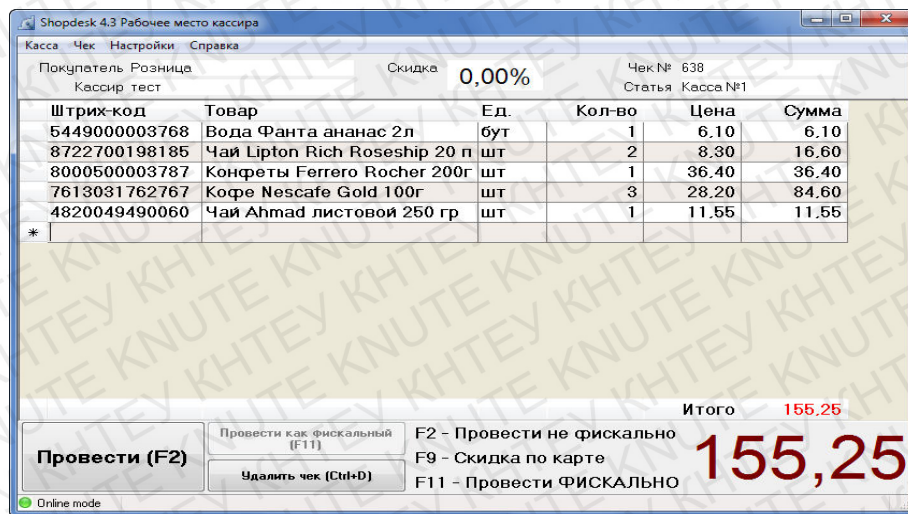


Рис. 1.2. – головне вікно програми ShopDesk

Програмна системи має низькі вимоги до апаратного та програмного обладнання, що дозволяє працювати з різноманітними типами та моделями сканерів та принтерів, не є винятком і чекові. Мінімізований час навчання користувачів та касирів даної програми. За допомогою зручного інтерфейсу касир має можливість ідентифікувати товар та внести його в рахунок за допомогою сканера штрих-кодів. Після сканування товару, немає значення чи він сталий чи ваговий, в рахунок записується «1» для сталого товару, тобто новий запис. В залежності від кількості касир може змінити дане значення на необхідну кількість. Якщо мова йде про ваговий товар, то відбувається сканування товару за допомогою штрих-коду, який раніше було надруковано на етикетці спеціальним принтером.

Програма SoftKuB

SoftKuB – інший аналог ІУС мінімаркету. OLAP-система для настільних ПК. Головним призначенням даної програми є аналіз даних х побудовою графіків у реальному часі. Це може бути графік побудований по різних критеріях, наприклад продажі по датах (місяцях, декадах, кварталах), також можуть бути різні групи товарів та агентів. Кількість зусиль для побудови графіку зводиться до мінімуму. Ключовою особливістю даної системи є швидкодія. Також одною з особливостей є сумісність з «1С Підприємство» та іншими подібними ПЗ. Підтримується також сумісність з реляційними БД. Що в свою чергу допомагає досягти високої ефективності бізнесу.

Одним з основних компонентів для відображення даних є Pivot Grid – інструмент, який дозволяє будувати графіки або «куби» звичайним перетягуванням мишею потрібної інформації, побудова відбувається в реальному часі.

Завдяки цій функції можна за декілька секунд з звіту продажі по датах утворити звіт продажі по інших критеріях, наприклад по клієнтах [14].

Програма дозволяє виділити мишкою певну ділянку графіка, яка необхідна.

Це дозволяє автоматично відобразити графік або діаграму у потрібному

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

вигляді. Саме ця особливість дозволяє, при потребі, максимально швидко скоротити час побудові графіка з необхідними перерізом даних. За допомогою OLE DB або ODBC можливо передати дані для побудови графіків. Це можуть бути як Excel файли так і бази даних різних форматів, наприклад MBD або бази даних типу SQL.

Для цієї програми на ринку є велика кількість конкурентів, однак перевагою саме SoftKub є ціна та зручність самої роботи з нею. Дане програмне забезпечення дозволяє будувати графіки за допомогою простого перетягування мишкою, що в свою чергу є унікально серед такого роду ПЗ. І не менш приємною є ціна, адже вона у три, чотири рази нижча і ніж у конкурентів, які можуть надати хоча б наближений функціонал.

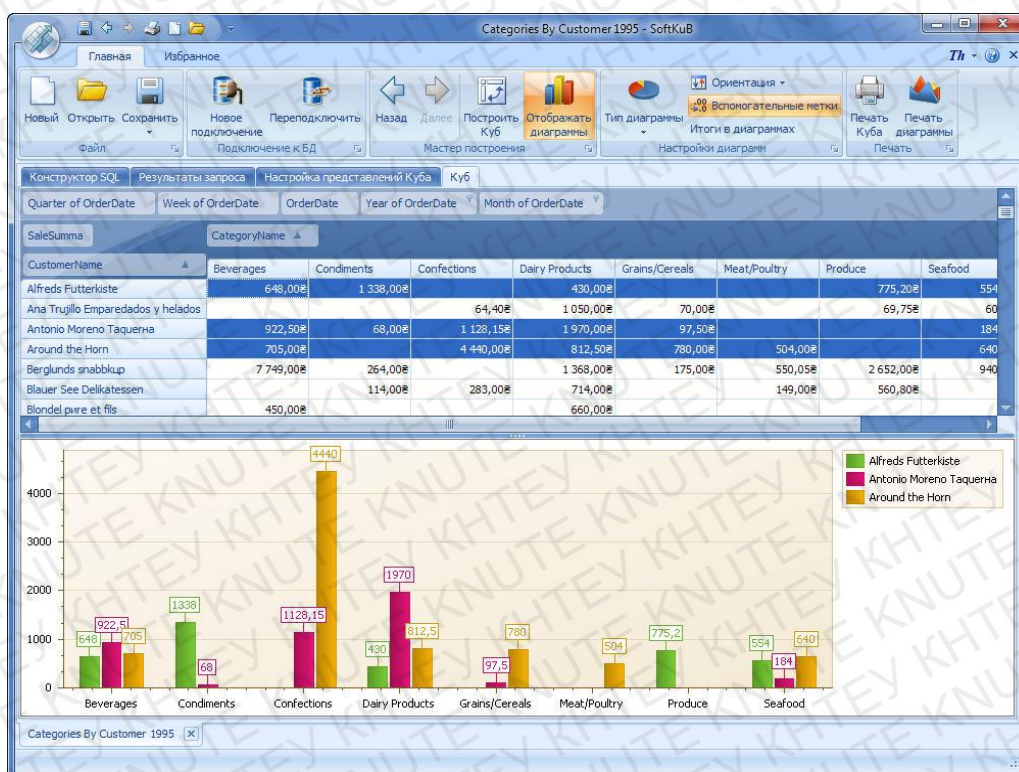


Рис. 1.3 – головне вікно програми SoftKub

Програма «Товари, Ціни, Облік..»

«Товари, Ціни, Облік...» – ще один з аналогів. Основне призначення програми – облік товару на виробництвах та у магазинах, середній бізнес.

Також програма дозволяє вести облік в торгових точках, складах, а також в

мінімаркетах. Дозволяє приймати оплати від клієнтів. Відстежує обсяг здійснених продаж. Листи розпорядження та їх маршрути. Розмежування прав користувачів системи. Дозволяю здійснювати контроль за операторами. В основному програму застосовують індивідуальні підприємства та супермаркети, роздрібна та оптова торгівля. Система працює в мережі з одною базою даних та має можливість підключати необмежену кількість інших користувачів, робочих місць.

«Товари, Ціни, Облік ...» надає широку можливість по розподілу прав у межах системи та організації. Кожному користувачеві надається рівно стільки прав у системі, скільки від нього вимагається для виконання максимально якісної роботи. Також є можливість налаштувати досить якісно та детально, що дозволяє утворювати певні зв'язки для взаємоконтролю працівникам один одного. Саме при умові такої роботи досягається максимальна ефективність роботи колективу. Окрім цього система веде журнал усіх операцій та дій зроблених операторами.

Також покращенню продуктивності сприяє максимально зручний та інтуїтивний інтерфейс, до якого досить швидко звикається.

					<i>КНТЕУ 121 023-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

Флаг	Вну тр. Ном	Номер	Расх / Прих	Торговая точка	Поставщик/Получатель	Дата проводки	Тип документа	Утвержде н	Сумма в закупочных	Сумма розничная	Сумма оплаты	Пл т с
	560814	22027	Расход	Склад №1	СПД Бубуренко Александр Анат	26.05.2009	Расх. накл.	Да	488,35	679,50	380,76	0,
	560815	22027	Приход	Склад №1	Склад №1	26.05.2009	Переоценка	Да		-54,30		0,
	560816	22028	Расход	Склад №1	СПД Черняк Сергей Валентинович	26.05.2009	Расх. накл.	Да	89,79	98,28	98,28	0,
	560817	22028	Приход	Склад №1	Склад №1	26.05.2009	Переоценка	Да		-4,11		0,
	560818	22029	Расход	Склад №1	СПД Шептур Наталья Владимир	26.05.2009	Расх. накл.	Да	37,55	40,70	40,70	0,
	560819	22029	Приход	Склад №1	Склад №1	26.05.2009	Переоценка	Да		-2,15		0,
	560820	22030	Расход	Склад №1	ТПП "Ромашка"	26.05.2009	Расх. накл.	Да	61,25	83,48	0,00	0,
	560821	22031	Расход	Склад №1	СПД Суетина Тамара Федоровн	26.05.2009	Расх. накл.	Да	617,38	791,70	82,08	0,
	560822	22032	Расход	Склад №1	ТПП "Ромашка"	26.05.2009	Расх. накл.	Да	59,86	61,44	0,00	0,
	560825	22033	Расход	Склад №1	СПД Вакаренко Галина Алексан	26.05.2009	Расх. накл.	Да	49,40	70,80	0,00	0,
	560826	22033	Приход	Склад №1	Склад №1	26.05.2009	Переоценка	Да		-6,61		0,
	560827	22034	Расход	Склад №1	СПД Чумаченко Надежда Васил	26.05.2009	Расх. накл.	Да	846,20	999,58	0,00	0,
	560850	19714	Приход	Склад №1	СПД Постовая Зинаида Михайл	26.05.2009	Касса (приход)	Да		151,00	0,00	0,
	560851	19715	Приход	Склад №1	СПД Терляк Юрий Ростиславов	26.05.2009	Касса (приход)	Да		426,79	426,79	0,
	560853	19716	Приход	Склад №1	СПД Лозовский Валерий Григор	26.05.2009	Касса (приход)	Да		28,80	28,80	0,
	560854	19717	Приход	Склад №1	СПД Лозовский Валерий Григор	26.05.2009	Касса (приход)	Да		405,72	405,72	0,
		17273			Средний чек: 351,85 грн.				1 200 010,06	6 077 513,20		

Рис. 1.4 – головне вікно програми «Товари, Ціни, Облік ...»

1.4. Вибір моделі розробки ІУС

Після глибокого аналізу вимог до системи та специфіки вибраної предметної області, було вирішено обрати еволюційну модель розробки ПЗ.

Саме завдяки цій методології з'являється можливість для глибокого дослідження предметної області, що в свою чергу дозволяє провести аналіз потреб замовника і придатність даної моделі в межах наявного проекту. Цю модель використовується в основному при розробці не критичних систем і головною метою є реалізація системного функціоналу. Також допускається, що вимоги до системи визначені не повністю. Саме через це реалізація ключового функціоналу відбувається ітераційно, що дозволяє на кожному етапі отримувати робочий прототип на якому власне і відбувається перевірка закладених раніше вимог та функціоналу. При дотриманні такого процесу стає можливим розробити максимально якісний продукт, який задовільнить усі вимоги навіть найвибагливішого замовника будь-якої системи. Дещо схожою є спіральна

						Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

модель розробки ПЗ [7].

Наступним кроком цієї моделі є еволюційне прототипування в межах розробки усього ЖЦ програмної системи (рис. 1.5).

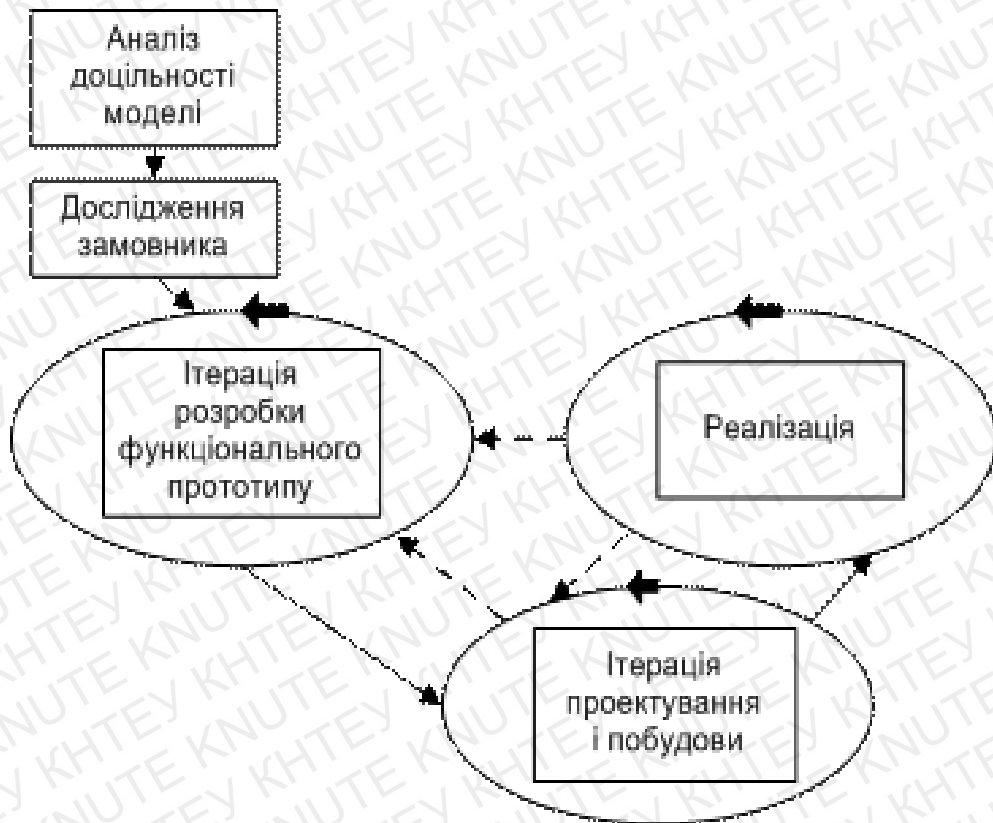


Рис. 1.6. Принцип роботи еволюційної моделі

Ця модель виконує дії, що дозволяють провести глибокий аналіз предметної області і застосувати його до конкретної системи, окрім цього є можливість підтримувати зв'язок з замовником, щоб більш детально визначити потреби користувачів про створенні прототипів на етапі ітерацій.

Дана модель має два головних етапи, або їх ще називають ітераціями. Починається все з проектування, яке далі перетікає в реалізацію, що в свою чергу дозволяє перевірити чи ПЗ задовольняє усі вимоги. Все зводиться до поступової реалізацію усього функціоналу через прототипи. Кожен наступний прототип доповнює попередній, що в свою чергу допомагає реалізувати увесь функціонал системи.

Переважно відбувається декілька ітерацій, під час яких створюється прототипи з додатковим функціоналом і заново моделюється робота. Це

відбувається до тих пір поки не буде протестовано та реалізовано увесь функціонал. Завершенням є фінальна ітерація – отримання готової версії продукту [6].

Дана модель має місце у системах з важливими функціональними можливостями, де потрібно максимально швидко продемонструвати функціонал.

Враховуючи те, що проміжні варіанти системи уже реалізують певний функціонал, виникає можливість їх перевірки, тестування та супроводу вже під час експлуатації даного прототипу, це означає, що разом з розробкою нової версії ПЗ відбувається тестування частини функціоналу попереднього прототипу.

Однак, варто враховувати наступні ризики:

- можливе нагромадження коду, даних та версій системи через одночасну реалізацію усіх функцій;
- даний метод є досить затратним в плані часу і потребує велику кількість людей та ресурсів, які в свою чергу будуть зайняті над проектом впродовж досить тривалого часу;
- в певних випадках дороговизна.

Дана модель має наступні переваги:

- висока швидкість реалізації частини функціоналу та їх тестування;
- можливість використовувати попередній прототип у наступній ітерації;
- відокремлення та реалізація частини функціоналу у вигляді готового прототипу;
- при потребі можна розширити систему про збільшенні фінансових надходжень;

Дана модель має наступні переваги:

- висока швидкість реалізації частини функціоналу та їх тестування;
- можливість використовувати попередній прототип у наступній

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

ітерації;

- відокремлення та реалізація частини функціоналу у вигляді готового прототипу;
- при потребі можна розширити систему про збільшенні фінансових надходжень;
- повторне та декілька разове узгодження вимог, без впливу на якість кінцевого продукту;
- можливість легше вносити зміни при реалізації окремих функцій.

Також для даної моделі ключову роль займає захист системи та даних, власне в цьому напрямку вона і розвивається [6].

1.5. Вибір технологій та інструментів для розробки

Мова програмування C#

Для написання магістерської роботи було обрано мову програмування C#. Саме ця мова ідеально підходить для розробки додатків на ПК і має для цього всі необхідні та зручні робочі інструменти.

C # - строго типізована об'єктно-орієнтована мова програмування. Вона являє собою відкритий код, простий, сучасний, гнучкий та універсальний.

Мова програмування - це мова, яка використовується для написання програмних програм.

C # - мова програмування, розроблена та запущена Microsoft у 2001 році, проста, сучасна та об'єктно-орієнтована мова, яка забезпечує сучасну розробку гнучкістю та можливостями для створення програмного забезпечення, яке не тільки працюватиме сьогодні, але застосовуватиметься роками у майбутньому. Вона є проста, сучасна та об'єктно-орієнтована мова програмування [1].

Ключові характеристики мови C # включають:

- Сучасна і легкий;

					КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

- Швидкий та відкритий код;
- Мова є кросплатформною;
- Безпечна;
- Універсальна;
- Постійно розвивається.

Якщо подивитися на історію мов програмування та їх особливості, кожна мова програмування була розроблена з певною метою, щоб вирішити конкретну потребу в той час.

Однак мова C # була створена для того, щоб вирішувати потреби бізнесу та підприємств. Мова була розроблена для створення всіх видів програмного забезпечення за допомогою однієї єдиної мови програмування.

Дана мова забезпечує функціональність для підтримки сучасної розробки програмного забезпечення. C # підтримує розробку веб, мобільних додатків та програм для ПК.

Деякі з сучасних функцій мови програмування підтримують дженеріки, типи var, автоматичну ініціалізацію типів і колекцій, лямбда- вирази, динамічне програмування, асинхронне програмування, кортежі, відповідність шаблонів, вдосконалене налагодження та обробку виняткових ситуацій та інше.

На синтаксис мови впливають C ++, Java, Pascal та кілька інших мов, які легко прийняти. C # також уникає складності та неструктурованих особливостей мови.

C # - безпечна мова типу, вона не дозволяє перетворення типів, які можуть призвести до втрати даних або інших проблем, саме це дозволяє розробникам писати безпечний код, а також зосереджується на написанні ефективного коду [8].

Також важливим є програмування між платформами. Можливо створювати .NET програми, які можна розгорнути на платформах Windows, Linux та Mac, а також можна розгорнути у хмарі та контейнерах.

Ось перелік деяких ключових моментів у C #, які допомагають писати

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

безпечний та ефективний код:

- Злиття небезпечних типів заборонене;
- Підтримка даних типу Null;
- Використовує повернення з типом «тільки для читання», що дозволяє використовувати копії об'єктів під час компіляції;
- Коли розмір структури, що зчитується більший, ніж IntPtr.Size, її слід передати її як параметр з міркувань продуктивності.
- Ніколи не передає структуру як параметр, якщо вона не оголошена модифікатором readonly, оскільки це може негативно вплинути на продуктивність і може призвести до незрозумілої поведінки.

Також дана мова є універсальною. Хоча більшість мов програмування розроблені з певною метою, C # був розроблений для майже всього. Можна використовувати для створення сучасних програмних систем, для розробки всіх видів програм, включаючи клієнтські, компоненти та бібліотек, сервіси та API, веб-додатки, мобільні додатки, хмарні програми та відеоігри [9].

Ось список типів додатків, які C # може скласти:

- Клієнтські програми Windows;
- Бібліотеки та компоненти Windows;
- Послуги Windows;
- Веб-додатки;
- Веб-сервіси та веб-API;
- Native iOS та мобільні додатки для Android;
- Сервіси з вихідного дня;
- Хмарні програми та сервіси Azure;
- Резервна база даних з використанням засобів ML / Data;
- Штучний інтелект та машинне навчання;

Фреймворк .Net

						КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			19

Фреймворк .Net - це платформа розробки програмного забезпечення, розроблена Microsoft. Фреймворк призначений для створення додатків, які працюватимуть на платформі Windows. Перша версія фреймворку .Net була випущена в 2002 році.

Версія отримала назву .Net Framework 1.0. З цього часу .Net пройшов довгий шлях, а поточна версія - 4.7.1.

Він може використовуватися для створення як програм на основі форм, так і на веб-основі. Веб-сервіси також можуть бути розроблені з його допомогою [1].

Основна архітектура .Net фреймворку наведена нижче на рисунку 1.7.

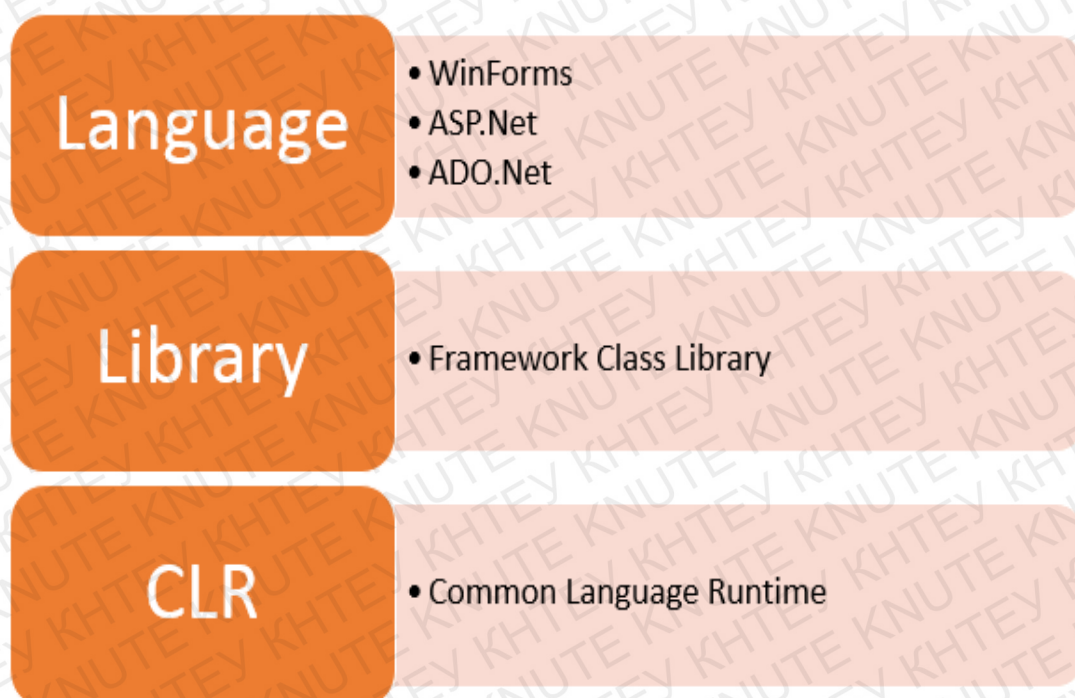


Рис. 1.7 – Основна архітектура .Net фреймворку

Архітектура .Net Framework базується на саме на цих ключових компонентах.

Common Language Infrastructure або "Загальна мовна інфраструктура", також відома як CLI - це платформа, на якій виконуються програми .Net.

CLI має такі основні особливості:

– Обробка винятків - Винятки - це помилки, які виникають під час

						КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			20

виконання програми.

Приклади винятків:

- Якщо програма намагається відкрити файл на локальній машині, але файл відсутній;
 - Якщо програма намагається отримати деякі записи з бази даних, але підключення до бази даних не дійсне.
- Garbage Collection – збирання сміття - це процес видалення небажаних ресурсів, коли вони більше не потрібні.

Приклади збору сміття є:

- Файл який більше непотрібен. Якщо програма закінчила всі операції над файлом, обробка файлів більше не потрібна;
 - Підключення до бази даних більше не потрібне. Якщо програма закінчила всі операції з базою даних, можливо, більше не потрібно буде з'єднання з базою даних і для оптимізації пам'яті воно закривається [8].
- Робота з різними мовами програмування.

Як зазначалося вище, розробник може розробляти додаток на різних мовах програмування .Net.

1. Мова – це перший рівень, тобто сама мова програмування, найпоширеніші - VB.Net та C #.
2. Компілятор - який буде окремим для кожної мови програмування. Отже, що лежить в основі мови VB.Net - це буде окремим компілятором VB.Net. Аналогічно, для C # передбачається інший компілятор який більш оптимізований під конкретну мову.
3. Інтерпретатор загальної мови - це останній рівень в .Net, який буде використовувався для запуску програми .net, розробленої на будь-якій мові програмування. Тож наступний компілятор відправить програму на рівень CLI для запуску програми .Net.

На рисунку 1.8 продемонстровано ключові компоненти CLI. Можна побачити, що спочатку відбувається оптимізація під конкретну мову

					<i>КНТЕУ 121 023-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		21

програмування з усіма її особливостями, а вже потім отримується доступ через загальний інтерпритатор [9].

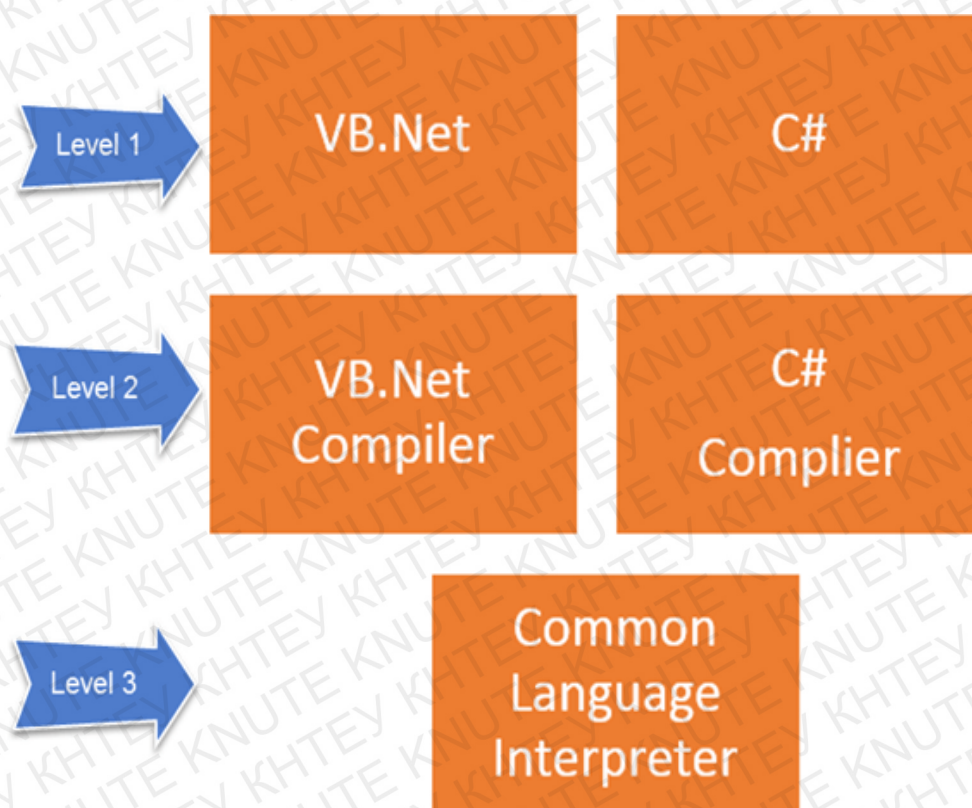


Рис. 1.8 – Особливості CLI

Class Library або Бібліотека класів. Окрім вище сказаного .NET Framework включає набір бібліотек стандартних класів. Бібліотека класів - це сукупність методів. Наприклад, є бібліотека класів з методами для обробки всіх операцій на рівні файлів.

Отже, існує метод, який можна використовувати для читання тексту з файлу. Так само існує метод запису тексту у файл.

Languages або мови. Типи програм, які можна вбудувати в .Net Framework, класифікуються в основному на наступні категорії:

- WinForms - використовується для розробки додатків на основі форм, які працюватимуть на машині кінцевого користувача. Блокнот - це приклад клієнтської програми;
- ADO.Net - Ця технологія використовується для розробки програм для

взаємодії з базами даних, такими як Oracle або Microsoft SQL Server;

- ASP.Net - використовується для розробки веб-додатків, які запускаються в будь-якому браузері, наприклад, Internet Explorer, Chrome або Firefox [1].

в та функцій, які можна використовувати для розробки.

Дану категорію умовно можна поділити ще на:

- Веб-додаток який оброблятиметься на сервері, на якому буде встановлено Інтернет-інформаційні послуги.

- Internet Information Services або IIS - це компонент Microsoft, який використовується для виконання програми Asp.Net.

- Результат виконання надсилається на клієнтські машини, а вихід відображається в браузері.

Microsoft завжди гарантує, що .Net співпрацює з усіма підтримуваними операційним системам Windows.

Окрім всього вище сказаного .Net Framework має певні принципи поведінки та дизайну.

Наступні принципи роблять даний фреймворк дуже актуальним для створення додатків на його базі:

- 1) Підтримка попередніх версій - .Net забезпечує підтримку попередніх версій платформи. Припустимо, якщо був додаток, побудований на старій версії, скажімо 2.0. Але якщо спробували запустити ту саму програму на машині, яка мала вищу версію фреймворку, 3.5 або вище наприклад, то додаток однаково запрацює. Це пов'язано з тим, що з кожним випуском Microsoft гарантує, що старі версії добре поєднуються з найновішою версією.

- 2) Портативність. Програми, побудовані на основі .Net, можна запустити на будь-якій платформі Windows. А останнім часом Microsoft також працює над тим, щоб їх продукти працювали на інших платформах, таких як iOS та Linux.

- 3) Управління пам'яттю - CLI виконує всю роботу по управлінню пам'яттю. Дана система має всі можливості бачити ті ресурси, які не

						КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			23

використовуються запущеною програмою. Тоді вони б відповідно звільнили ці ресурси. Це робиться за допомогою програми під назвою "Збір сміття", яка працює як частина загально механізму та забезпечує цілісність роботи.

4) Безпека – даний фреймворк має хороший механізм захисту. Вбудований механізм допомагає як при верифікації, так і для перевірки програм. Кожна програма може чітко визначити свій механізм захисту. Кожен механізм захисту використовується для надання користувачеві доступу до коду або до запущеної програми.

5) Спрощене розгортання – окрім цього передбачені інструменти, які можна використовувати для упаковки програм, побудованих на даній платформі. Потім ці файли можуть бути розповсюджені на клієнтських машинах, що дозволяю спростити їх встановлення.

Підсумовуючи вище сказане, .Net - фреймворк, створений Microsoft. Він був розроблений для створення додатків, які могли би працювати на платформі Windows. Дана платформа .Net може використовуватися для розробки додатків на Windows forms, веб-додатків та веб-служб. Розробники можуть вибирати з різних мов програмування, доступних на платформі .Net. VB.Net і C# найпоширеніші з них [8].

1.6. Середовище розробки Visual Studio

Інтегроване середовище розробки Visual Studio - це креативний інструмент, який можна використовувати для редагування, налагодження, створення програмних продуктів та їх публікації. Інтегроване середовище розробки (IDE) - це багатофункціональна програма, яка може бути використана для різноманітних напрямків розробки програмного забезпечення. Окрім стандартного редактора та налагоджувача, який надає більшість IDE, тут ще включені компілятори, інструменти для форматування та доповнення коду, графічні середовища та інші функції які значно полегшують процес розробки програмного забезпечення [5].

					КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24

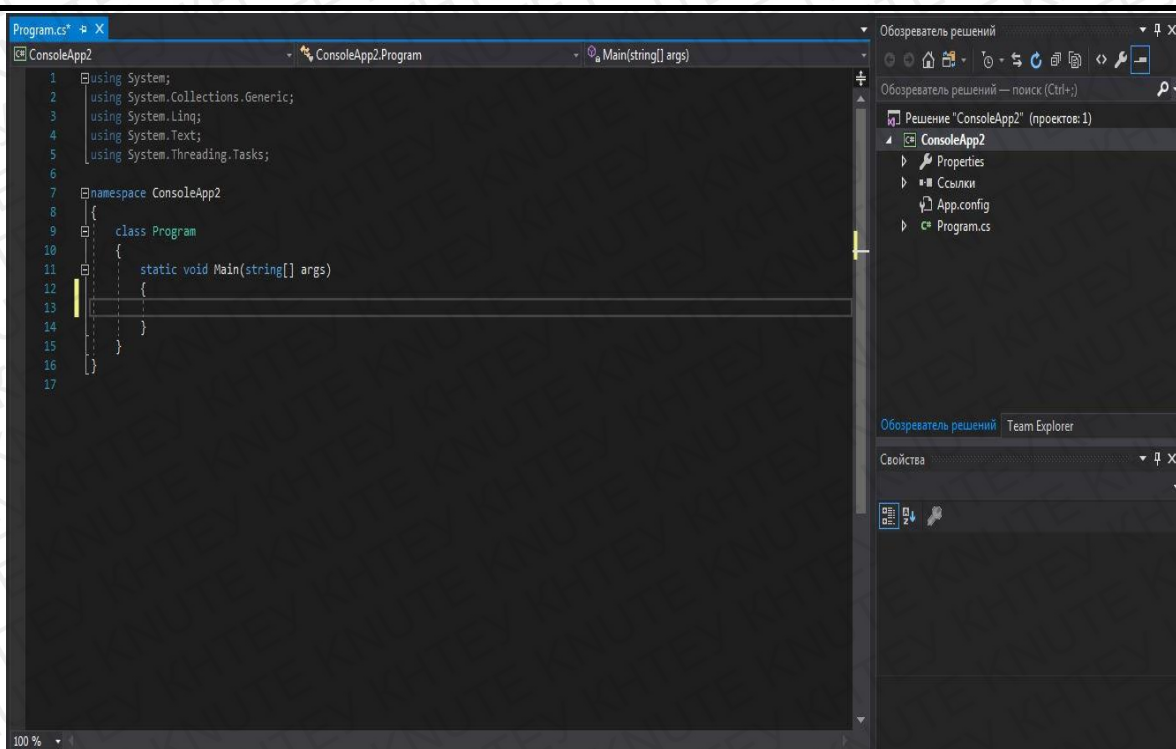


Рис. 1.9 Середовище розробки Visual Studio

На рисунку 1.9 зображено Visual Studio з відкритим проектом та кількома ключовими вікнами інструментів, Solution Explorer (угорі праворуч) дозволяє переглядати, переміщуватися та керувати файлами коду. Провідник рішень може допомогти впорядкувати код, згрупувавши файли у рішення та проекти.

- Editor window (в центрі) місце де проводиться більша частина свого часу, відображається вміст файлу.
- Team Explorer (внизу праворуч) дозволяє відстежувати робочі елементи та ділитися кодом з іншими за допомогою технологій контролю версій, таких як Git та Team Foundation (TFVC).

Visual Studio доступний для Windows та Mac. Visual Studio для Mac має багато тих же функцій, що й для Windows і оптимізований для розробки кросплатформних та мобільних додатків.

						КНТЕУ 121 02з-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			25

1.7. Висновки до Розділу 1

Сучасний магазин представляє собою великий вибір товар різних груп та категорій де покупець має можливість знайти все, що його цікавить в одному місці. Враховуючи кількість пропозицій товарів у магазинах та потребу в акційних продажах для підвищення прибутку, можна зробити висновок, що не всі системи задовільняють потреби сучасного покупця. Існуючі системи досить часто не пристосовані до великої кількості товарів та постійних цінових змін в середині системи. Також у великих маркетах не завжди вчасно відбувається зміна цінників. Проте за допомогою сучасних комунікаційних систем та мереж появляється безліч можливостей для вдосконалення існуючих систем та впровадження нових технологій.

Окрім вище сказаного, існує можливість спільно редагувати та налагоджувати налаштування з іншими користувачами в режимі реального часу, незалежно від типу обраного додатка чи мови програмування. Присутня можливість миттєво та надійно ділитися своїм проектом і при потребі проводити налагодження сесій, веб-додатків, голосових дзвінків та іншого.

Враховуючи все описане вище, дане середовище ідеально підходить для розробки додатку на мові програмування C#

					КНТЕУ 121 023-15.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

РОЗДІЛ 2

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЮЮЧОЇ СИСТЕМИ

2.1 Виявлення ризиків при проектуванні ІУС

Розробка будь-якої системи, незалежно від складності, несе в собі певні ризики, які завжди потрібно враховувати як розробнику і замовнику [14]. В таблиці 2.1. описані ключові ризики розробки ІУС мінімаркету.

Таблиця 2.1

Ризики при розробці ПЗ

Категорії ризиків	Ризики та їх приклади
1	2
Ризики технологічного характеру	Несправності апаратного забезпечення робочих машин персоналу; Збій у роботі бази даних при обробці запитів від користувачів; Не можливе повторне використання раніше розроблених модулів чи компонент в подібних задачах.
Ризики організаційного характеру	Не правильно сформований бюджет для розробки проекту; Перед тим як почати розробку не були остаточно ухвалені усі потреби і побажання замовника; Замовник змінює орієнтацію проекту в іншу сферу. Не можливість знайти хостинг для сайту.

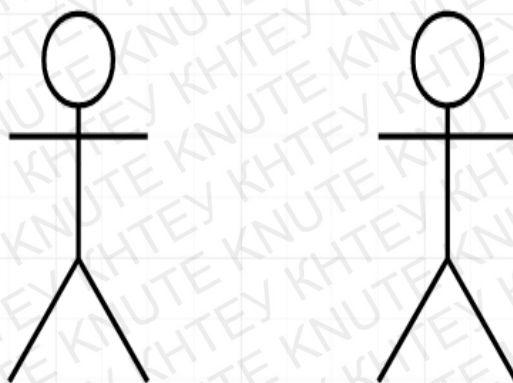
					<i>КНТЕУ 121 023-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		24.05.21	Проектування інформаційно-управляючої системи мінімаркету	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		24.05.21		P2	27	63
Гарант		Токар В.В.		24.05.21	Проектування інформаційно-управляючої системи	Факультет інформаційних технологій 23 курс, 2м група		
Розробив		Тищенко Д.О.		24.05.21				

<p>Персонал та ризики пов'язані з його наймом</p>	<p>Кількість набраного персоналу є малою для того, щоб розпочати реалізацію поставленого проекту; Труднощі у підборі висококваліфікованих працівників; Основні технології побудови програмної системи замінюються новими Неможливо реалізувати навчання персоналу.</p>
<p>Бізнес-Ризик</p>	<p>На ринку програмних продуктів до закінчення проекту з'явилася конкуруюча програмна система</p>

Продовження таблиці 2.1

2.2. Пошук акторів системи

Розробка будь-якої системи вимагає детального аналізу та розуміння не лише програмних засобів, але і предметної області в якій ведеться розробка [10]. Проаналізувавши предметну область та усі функціональні, нефункціональні вимоги, можна виділи двох чітких акторів в системі. Цими акторами є касир та адміністратор, рисунок 2.1.



Касир Адміністратор

					<p>КНТЕУ 121 023-15.МР</p>	<p>Аркуш</p>
<p>Зм.</p>	<p>Аркуш</p>	<p>№ докум</p>	<p>Підпис</p>	<p>Дата</p>		<p>28</p>

Рис. 2.1. Головні актори системи

Саме ці актори виступають головними користувачами АСУ мінімаркетом. В таблиці 2.2. наведено ключових акторів та описано їх головну роль в системі.

Таблиця 2.2.

Виявлення акторів

Актор	Короткий опис
1	2
Касир	Головний користувач програми який має змогу виконувати пошук товарів в системі по штрих-кодів або по звичайному цифровому кодів і додавати ці товари в чек. Основною його задачею є обслуговування клієнтів. Окрім цього також може виконувати в системі пошук персоналізованих дисконтних карток.
Адміністратор	Головний користувач програми який має змогу керувати даними в системі, а також додавати та видаляти в системі користувачів, відвідаю за надання прав та отримує доступ до звітності.

2.3. Виявлення основних варіантів використання

Окрім визначення акторів в системі, також необхідно встановити варіанти використання для кожного окремого користувача. Саме це описано в таблиці 2.3.

Таблиця 2.3

Опис варіантів використання

1	Касир	Авторизація в системі	Касир має змогу авторизуватися в системі для початку роботи з нею, а також завершити сеанс.
2	Касир	Застосувати дисконтну карту	Під час продажу товарів, касир має змогу застосувати дисконтну карту клієнта, при її наявності.

						КНТЕУ 121 02з-15.МР	Аркуш
							29
Зм.	Аркуш	№ докум	Підпис	Дата			

3	Касир	Змінити кількість товарів в чеку	Якщо товар відскановано і необхідно змінити його кількість в чеку, касир редагує дані в графі «Кількість» без потреби повторного сканування.
K4	Касир	Пошук товару в системі	При необхідності, користувач може знайти необхідний товар в системі за певними критеріями
K5	Касир	Додати товар в чек	Користувач додає товар в чек. Це можна зробити декількома способами: відсканувати штрих-код, знайти товар по назві або по номері.
K6	Касир	Додати товар в систему	При наявності доступу, користувач може вносити новий товар в систему при необхідності.
A1	Адміністратор	Адміністрування користувачів	Адміністратор може додати, видалити, редагувати дані про користувачів.
A2	Адміністратор	Робота з звітністю	Доступ до даних по продажі товарів.

Аналіз представлених варіантів використання встановив наступні взаємозв'язки.

Варіанти використання для касира «Додати товар в чек за допомогою штрих-коду», «Додати товар в чек за допомогою коду» та «Додати товар в чек по назві» можна узагальнити до «Додати товар в чек». Це пов'язано з тим, що по великому рахунку виконується одна і та сама дія, користувач

«Касир» шукає в системі товар, для того, щоб потім додати його в чек. На етапі а проектування не є критичним, чи це буде зроблено за допомогою сканера, чи за допомогою вводу коду товару в системі [12]. Результат досягається один і

						КНТЕУ 121 02з-15.МР	Аркуш
							30
Зм.	Аркуш	№ докум	Підпис	Дата			

той самий товар додається в чек, узагальнення представлено на рисунку 2.2.

					<i>КНТЕУ 121 023-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		31

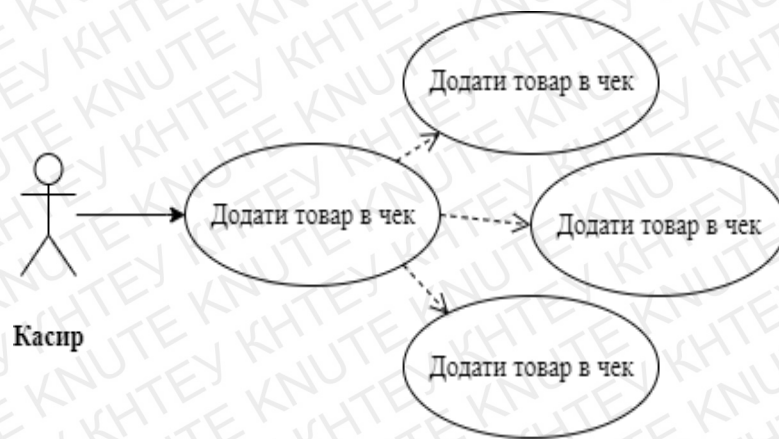


Рис. 2.2 Узагальнення варіантів «Додати товар в чек»

Аналогічна ситуація виникає і для варіанту використання «Застосувати дисконтну карту». Оскільки варіанти використання «Застосувати карту за кодом» та «Застосувати карту по штрих-коду» в плані логіки є однаковими, вони лише дозволяють отримати дані про дисконтну карту користувача. Це означає, що ці варіанти можна узагальнити до одного варіанту використання «Застосувати дисконтну карту». Це дозволить наглядно спростити представлення про систему, рисунок 2.3.



Рис. 2.3 Узагальнення пов'язане з дисконтною картою

Варто також звернути увагу на узагальнення «Адміністрування користувачів». Сюди входять усі варіанти пов'язані з керуванням користувачами системи. До них можна віднести «Додати користувача», «Видалити користувача», «Редагувати дані», «Зміна прав доступу».

Адміністратор це ключовий користувач який відповідає за цілісність та коректність роботи системи. Одним з основних його завдань є менеджмент нових користувачів. Також варто додати, що саме адміністратор керує правами доступу. Мається на увазі, що касири поділені на дві групи. Перша з яких має доступ лише до продажу товарів і може лише додавати товари в чек та продавати їх. Інша категорія має доступ на внесення нових товарів в систему. За допомогою сканера штрих-кодів та простого вводу коду товару, ці користувачі можуть додавати нові товари в систему [16]. Узагальнення зображене на рисунку 2.4. Адміністратор може як надавати ці права так і забирати їх, ніякої попередньої підготовки для цього не потрібно. Варто також зазначити, що передбачено обмеження у розмірі 30 касирів для даної системи. Оскільки наявність більшої кількості користувачів може значною мірою сповільнити систему, що призведе до некоректного функціонування. Окрім цього адміністратор має можливості, які не зазначені у варіантах використання. Наприклад це моніторинг за часом, який проводять за робочим місцем інші користувачів системи. Адміністратор може бачити сумарну кількість годин, які проводили касири за робочим місцем .



Рис. 2.4 Узагальнення пов'язане з варіантом використання «Адміністрування користувачів»

Наступне узагальнення це «Робота з звітністю». Даний варіант використання включає в себе такі можливості як «Звітність за відрізок часу», «Вивести звітність на екран», «Зберегти у форматі PDF». Даний варіант використання описує логіку пов'язану з переглядом даних про продажі магазину. Доступ до цих даних має лише адміністратор, рисунок 2.5.



Рис. 2.5 Узагальнення пов'язане з варіантом використання «Робота з звітністю»

Аналіз даних варіантів використання показав, що з погляду потенційних ризиків і архітектурної значущості найбільш істотними є прецеденти, пов'язані з певними варіантами використання акторів «Касир» та «Адміністратор». Певні прецеденти є ключовими в системі, адже саме на них зав'язана ключова логіка. Тому варто детальніше розглянути їх, саме ці користувачі користуються системою і певні варіанти використання мають найбільше значення. Варто крім цього звернути увагу, що для можливості доступу до певного варіанта використання потрібно зробити дії, щоб отримати доступ до усього функціоналу. В залежності від прав доступу,

«Касир» може лише отримувати можливість для додавання товару в чек та застосування дисконтних карток, а також може мати права, щоб додавати новий товар в систему, варто розмежовувати ці два моменти, адже від цих прав, задачі даної групи користувачів можуть відрізнитися.

Для подальшої деталізації вибрано такі прецеденти:

- K1. Авторизація в системі;
- K2. Застосувати дисконтну карту;
- K5. Додати товар в чек;
- A2. Робота з звітністю.

В таблиці 2.4 описаний варіант використання «Авторизація в системі». При у спішному завершенні прецеденту користувач отримує змогу надалі працювати з системою, якщо за певних причин логін або пароль були введені неправильно, у доступі до системи буде відмовлено. Для того, щоб відновити пароль або доступ до системи, користувач повинен зв'язатися з адміністратором та повідомити про неможливість входу в систему. Дану проблему адміністратор вирішує на свій розсуд. Можливий варіант, що касирові повідомлять старий пароль з допомогою якого відновиться можливість авторизації [6]. Інший варіант, це заміна паролю на новий. Примітка, в самій системі відсутні інструменти, механізми для користувач заміни користувачем даних для авторизації, тобто логіна або пароля.

Таблиця 2.4

Опис варіанту використання «Авторизація в системі»

Прецедент: Авторизація в системі	
ID:	K1
Короткий опис:	Касир має змогу авторизуватися в системі для початку роботи з нею, а також завершити сеанс.
Головні актори:	Касир
Другорядні актори:	Немає
Передумови:	Користувач зареєстрований в системі
Основний потік:	<ol style="list-style-type: none"> 1. Касир вмикає програму АСУ мінімаркетом 2. Після увімкнення з'являється поле для вводу логіна та пароля, відбувається процес авторизації.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		35

1. Користувач вводить логін та пароль у відповідні поля.
2. Система авторизує користувача.
3. Система відмічає в базі даних час початку роботи користувача.
4. Запускається головний інтерфейс для роботи з програмою.

Постумови:

При успішному закінченні прецеденту робляться відмітки в базі даних про час початку роботи користувача

Альтернативні потоки:

- 1.1 Невірно вказаний логін або пароль

Якщо при виконанні п. 3 було введено невірно логін або пароль, система видає повідомлення про це в діалоговому вікні.

Наступний важливий прецедент, це «Застосувати дисконтну карту», таблиця 2.5. При наявності дисконтної карти клієнт може отримати знижку. Для цього необхідно надати касиру цю карту, після чого він просканує її і система автоматично вирахує знижку. Якщо карту не вдається просканувати, завжди можна ввести її код, який знаходиться на самій карті. Картки додає в систему касир з відповідними правами доступу.

Таблиця. 2.5

Опис варіанту «Застосувати дисконтну карту»

Прецедент: Застосувати дисконтну карту	
ID: K2	
Короткий опис: Підчас продажу товарів, касир маю змогу застосувати дисконтну карту клієнта, при її наявності.	
Головні актори: Касир	
Другорядні актори: Немає	
Передумови: Користувач зареєстрований та авторизований в системи, покупець купляє принаймні один або більше товарів і вони внесені в чек.	
Основний потік: 1. Касир вмикає програму АСУ мінімаркетом 2. Касир авторизується в системі.	

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		36

3. Додається товар в чек за допомогою сканера штрих-кодів.
4. Натискається на кнопку, що дозволяє оформити знижку по дисконтній картці.
5. Відкривається діалогове вікно.
6. За допомогою сканера штрих-кодів касир сканує карту.
7. Вираховується нова загальна сума чеку з врахуванням знижки.

Постумови:

При успішному закінченні прецеденту вираховується нова ціна в чеку з врахуванням знижки.

Альтернативні потоки:

1.1 Невірно вказаний логін або пароль

Якщо при виконанні п. 3 було введено невірно логін або пароль, система видає повідомлення про це в діалоговому вікні.

1.2 Не знайдено карти

Якщо при виконанні п. 6 не вдалося розпізнати карту, касир може ввести вручну номер карти.

1.3 Жодного товару не додано в чек

Якщо при виконанні п. 4 в списку товарів, в чеку, не було жодного товару, система видасть повідомлення, що необхідно додати товар.

«Додати товар в чек» - один з ключових варіантів використання, описаний в таблиці 2.6. Даний прецедент відіграє ключову роль в цілій системі, адже він відповідає за додавання товару в чек. У разі якщо не вдається просканувати штрих-код або він пошкоджений, користувач може ввести код товару та додати його в чек. Також в системі можуть бути товари без штрих-коду. В такому випадку додати їх в чек можливо лише за наявності коду товару. Якщо один товар було проскановано декілька раз, позначка в графі «Кількість» буде рівна кількості сканувань.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

Опис варіанту « Додати товар в чек »

Прецедент: Додати товар в чек	
ID: K5	
Короткий опис: Касир сканує товар та додає його в чек	
Головні актори: Касир	
Другорядні актори: Немає	
Передумови: Користувач зареєстрований та авторизований в системі, покупець купляє принаймні один або більше товарів.	
Основний потік: <ol style="list-style-type: none"> 1. Касир вмикає програму АСУ мінімаркетом 2. Касир авторизується в системі. 3. Додається товар в чек за допомогою сканера штрих-кодів. 4. Товар появляється в таблиці, яка відображає список усього чеку. 5. При необхідності касир змінює кількість конкретного товару в таблиці. 6. Вираховується загальна сума чеку. 	
Постумови: При успішному закінченні прецеденту товар додається в чек, а також вираховується загальна сума чеку	
Альтернативні потоки: <ol style="list-style-type: none"> 1.1 Невірно вказаний логін або пароль Якщо при виконанні п. 3 було введено невірно логін або пароль, система видає повідомлення про це в діалоговому вікні. 1.2 Не знайдено товару Якщо при виконанні п. 3 не вдалося розпізнати товар, касир може ввести вручну номер товару. 1.3 Жодного товару не додано в чек Якщо не було додано жодного товару в чек, система видає повідомлення про необхідність додати принаймні одного товару для закриття чеку. 	

Також адміністратор має можливість працювати з інформацією по продажах, описано в таблиці 2.7. За допомогою цього може отримати доступ до інформації пов'язаною з продажами за обраний період часу. Можна перегляну

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		38

який саме товар, та в якій кількості було продано. При необхідності цю інформацію можна зберегти на власний комп'ютер. За допомогою цієї інформації можна провести детальний аналіз діяльності магазину та усіх продажів і при необхідності провести оптимізаційні кроки для покращення роботи підприємства [14].

Таблиця 2.7

Опис варіанту використання « Робота з звітністю »

Прецедент: Робота з звітністю	
ID: A2	
Короткий опис: Доступ до даних по продажі товарів. Присутня можливість роздрукувати або вивести на екран усі дані.	
Головні актори: Адміністратор	
Другорядні актори: Немає	
Передумови: Користувач зареєстрований та авторизований в системи, було здійснено принаймні один продаж	
Основний потік: <ol style="list-style-type: none"> 1. Адміністратор вмикає програму АСУ мінімаркетом 2. Авторизується в системі. 3. Вибирає вкладку «Звітність». 4. Встановлю проміжок за який потрібно отримати дані по звітності. 5. Система виводить інформацію по звітності на екран. 6. При необхідності, адміністратор зберігає її на комп'ютер. 	
Постумови: При успішному закінченні прецеденту адміністратор отримує на своєю ПК документ з звітністю за обраний період	
Альтернативні потоки: <ol style="list-style-type: none"> 1.1 Невірно вказаний логін або пароль Якщо при виконанні п. 3 було введено невірно логін або пароль, система видає повідомлення про це в діалоговому вікні. 1.2 Не знайдено звітності Якщо при виконанні п. 4 системі не вдалося нічого знайти або виникла помилка, користувач отримує повідомлення про це. 	

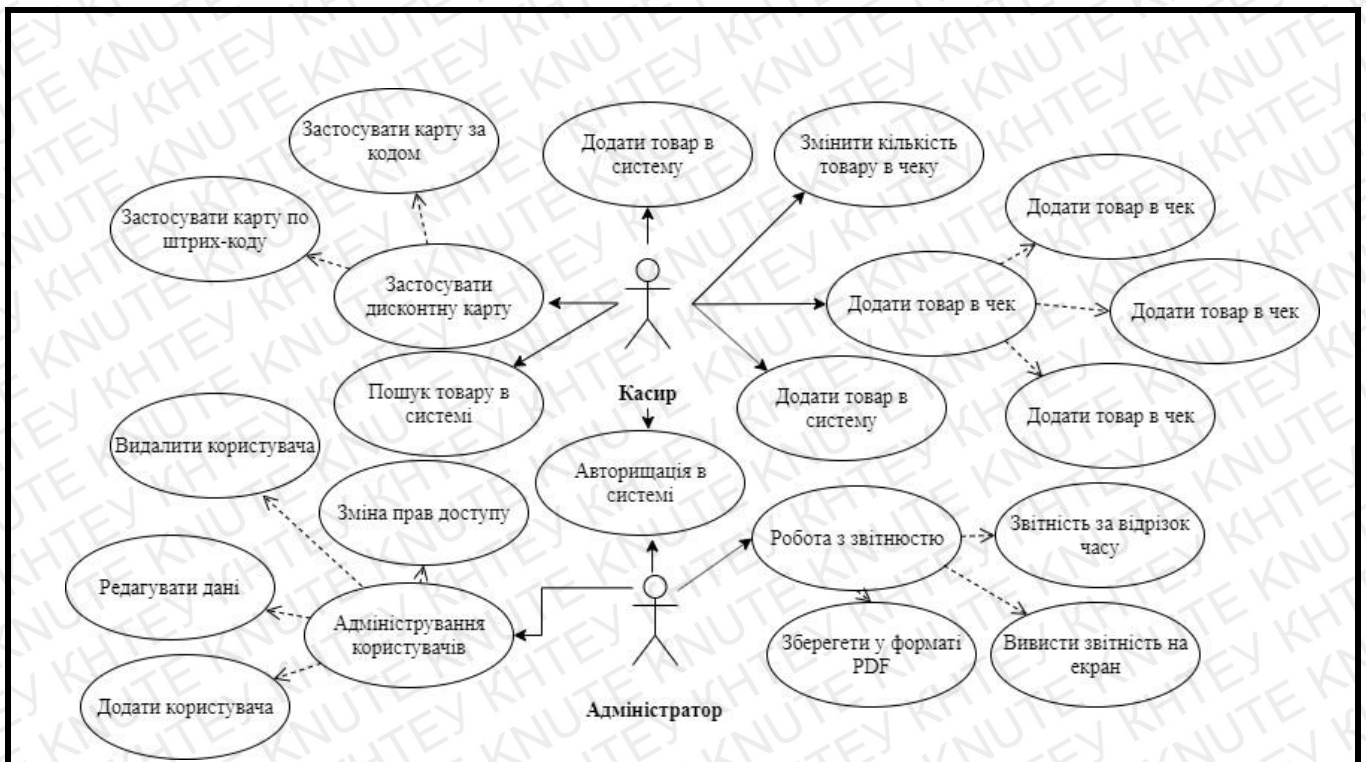


Рис. 2.7 Ключові варіанти використання

На рисунку 2.7 описано ключові варіанти використання. Частина з них вже була описана вище, розглянемо решту. «Адміністрування користувачів» - один з прецедентів який забезпечує функціонал пов'язаний з додаванням інших користувачів в систему. Дозволяє адміністратору додати в систему нового користувача. А також при необхідності видалити вже існуючого. Також ключовою функцією є надання прав доступу вже наявним користувачам. Завдяки цьому можна керувати правами доступу вже наявних користувачів та надавати або забирати в них частину функціональних можливостей. «Пошук товару в систему» - забезпечує касиру можливість знайти наявний товар в системі. Це можна зробити за допомогою штрих-коду або просто ввівши частину назви цього товару в стрічку для пошуку. Це дозволяє перевірити його наявність та ціну. Також важливу роль відіграє

«Додати товар в систему». Цією функцією можуть скористатися лише користувачі з відповідними правами доступу, які надає адміністратор. За

						Аркуш
					<i>КНТЕУ 121 02з-15.МР</i>	40
Зм.	Аркуш	№ докум	Підпис	Дата		

допомогою цієї функції касир може додавати новий товар в систему. Для цього необхідно натисну «Додати товар». Після чого появиться відповідна форма для заповнення. У цій формі необхідно вказати назву товару, код для ідентифікації, а також відсканувати штрих-код [6].

2.4. Проектування діаграми послідовності

Визначивши усі варіанти використання, можна описати саму роботу системи більш детально. Щоб це описати використаємо один з різновидів діаграми UML, а саме діаграму послідовності. Ця діаграма дає змогу краще зрозуміти взаємодію об'єктів, які впорядковані за часовим критерієм. Ця діаграма найкраще описує, які об'єкти були задіяні та в якій послідовності відбувався обмін повідомленнями між ними. В основному опис відбувається за допомогою горизонтальних та вертикальних ліній. Де вертикальні – це процеси чи об'єкти, а в свою чергу горизонтальні – це повідомлення які були надіслані. Ці повідомлення, як правило впорядковані за відправленням [6].

Переважно на цій діаграмі показують лише ті об'єкти, що беруть участь у обміні інформацією, можливі або статичні об'єкти не враховуються. Для діаграми даного типу є важливою послідовність об'єктів та динаміка з якою вони взаємодіють [7].

На рисунку 2.6 представлена діаграма послідовності для варіанту використання «Додати товар в чек». Спочатку касир дає запит на пошук товару, за допомогою сканування штрих-коду, далі система шукає в базі даних відповідний товар.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

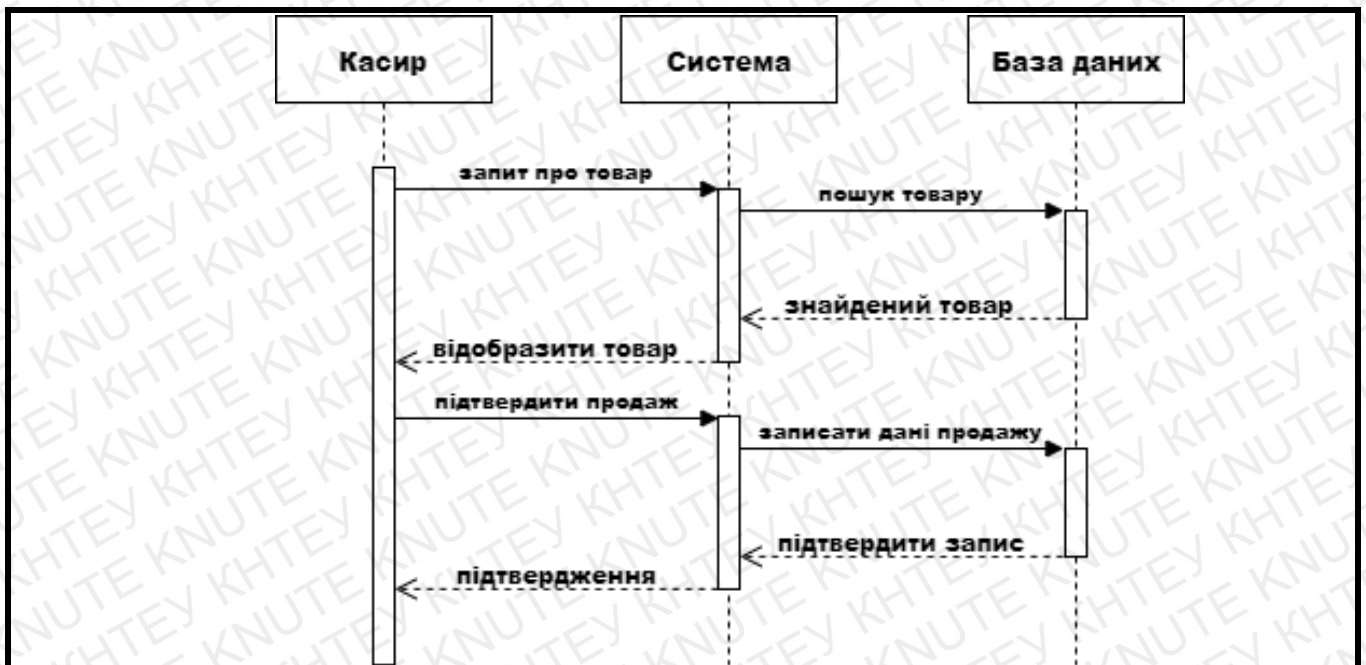


Рис. 2.6 Діаграма послідовності «Додати товар в чек»

Також варто розглянути варіант використання «Робота з звітністю». Роботу починає адміністратор, вибираючи вкладку «Інформація про звітність». Після чого відправляється відповідний запит в систему. Коли система отримала та опрацювала запит, він передається далі у базу даних. В залежності від обраних критеріїв виконується пошук даних. Знайдені дані назад повертаються в систему, яка опрацьовує їх відповідним чином та відображає користувачеві у табличному вигляді. Після отримання інформації користувач має вибір, завершити перегляд інформації про звітність або зберегти її на комп'ютер.

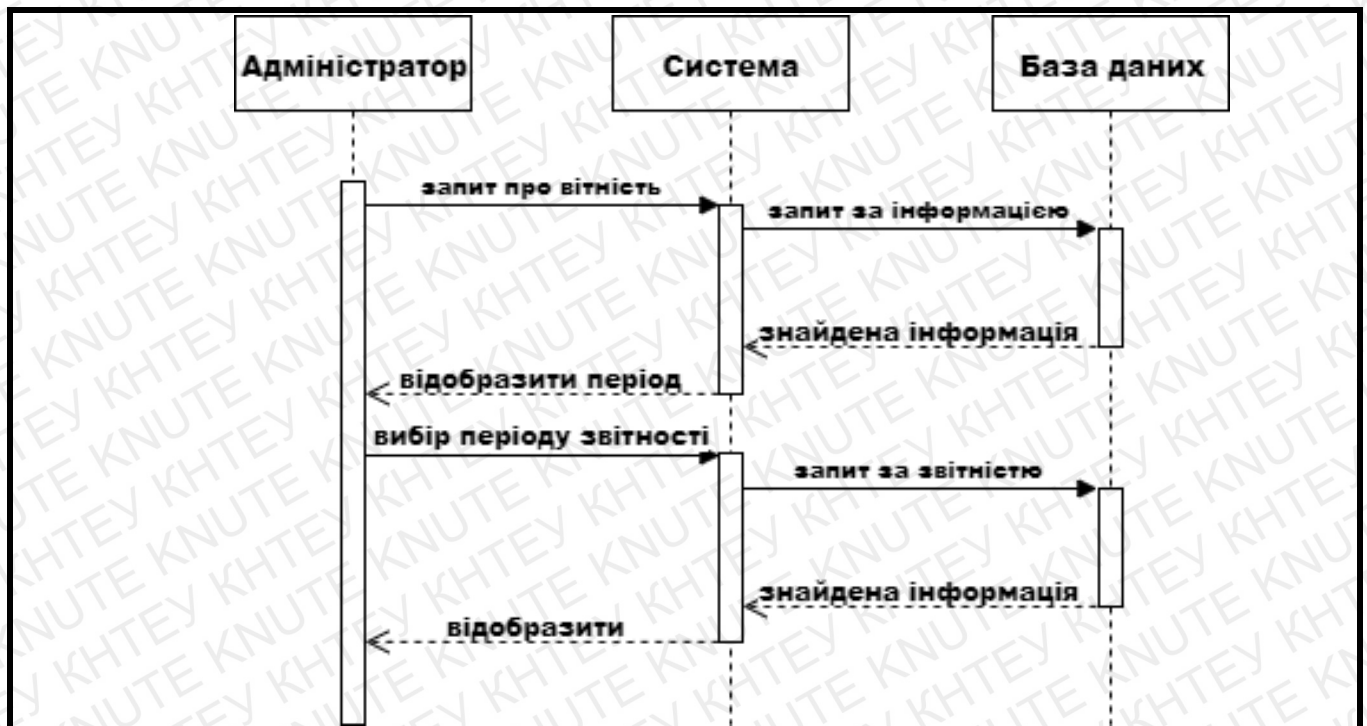


Рис. 2.7 Діаграма послідовності «Інформація про звітність»

Дана діаграма дозволяє поглибити розуміння роботи окремих елементів програми та спростити конструювання та розробку ПЗ. Окрім головних сценаріїв використання, варто також пам'ятати про альтернативні потоки, які можуть значною мірою ускладнити проектування та реалізацію. При роботі з таким типом потоків завжди потрібно пам'ятати про необхідність якщо не всіх, то практично всіх альтернативних сценаріїв, адже навіть одне недоопрацювання в котромусь з напрямків може привести до критичної помилки та краху усієї системи. Тому також варто пам'ятати про обробку помилок в системі [6].

2.5. UML-діаграми активності

Для більш детального розуміння ВВ «Застосувати дисконтну карту» представимо його за допомогою діаграми активності, рисунок 2.8. Ця діаграма дозволяє покроково відтворити потрібний нам сценарій з наглядною демонстрацією усіх кроків та етапів виконання даної операції.. Діаграма

						Аркуш
					<i>КНТЕУ 121 02з-15.МР</i>	43
Зм.	Аркуш	№ докум	Підпис	Дата		

активності починається з стартової позиції, з якої отримуються початкові дані і продовжується робота з ними. Після чого події можуть розвиватися поступово або паралельно. Також можливі розгалуження тоді, коли можливий альтернативний сценарій.



Рис. 2.8 Діаграма активності «Застосувати дисконтну карту»

Дана діаграма починається з отримання запиту на сканування карти. Після чого касир сканує дисконтну картку клієнта. Якщо карту не вдається знайти касир може ввести дані карти вручну. Якщо ж знову карти не знаходить, в діалоговому вікні відображається інформація про те, що не вдалося знайти такої карти. Коли карту вдалося просканувати номер цієї карти передається в базу даних для пошуку точної інформації про суму знижки [17]. Після того як інформацію було знайдено база даних повертає її назад в систему, яка в свою чергу розраховує нову ціну вже з врахуванням знижки. Після чого на екрані касира відображається остаточна ціна з врахуванням знижки.

Діаграма активності визначається як діаграма UML, яка фокусується на виконанні та потоці поведінки системи замість реалізації. Її також називають об'єктно-орієнтованою блок-схемою. Діаграми діяльності складаються з заходів,

що складаються з дій, які стосуються технології поведінкового моделювання.

Вона дозволяє створювати події як діяльність, що містить набір вузлів, з'єднаних ребрами. Діяльність може бути приєднана до будь-якого елемента моделювання, щоб змоделювати її поведінки. Діаграми активності використовуються для моделювання:

- Варіантів використання;
- Класів;
- Інтерфейсів;
- Компонентів;
- Взаємодії.

Діаграми активності використовуються для моделювання процесів та робочих потоків. Суть діаграми зосереджена на повідомленні конкретного аспекту динамічної поведінки системи. Вони фіксують свою увагу на динамічних елементах.

Вона схожа на блок-схему, яка візуалізує потік від однієї діяльності до іншої діяльності. Крім того, може бути ідентична блок-схемі, але вона не є блок-схемою. Потік активності можна керувати за допомогою різних елементів керування на діаграмі UML. Простими словами, діаграма використовується для потоків діяльності, які описують потік виконання між декількома видами діяльності [6].

Щоб скласти діаграму такого типу, потрібно зрозуміти та дослідити всю систему. Користувачі повинні знати всі елементи та об'єкти, які будуть використовуватися всередині діаграми. Користувачеві має бути зрозуміло центральне поняття, яке є не що інше, як принцип роботи системи. Проаналізувавши всі види діяльності, щоб знайти різні обмеження, що застосовуються до реалізації. Якщо існує таке обмеження, то це слід зазначити перед розробкою.

						Аркуш
					<i>КНТЕУ 121 023-15.МР</i>	45
Зм.	Аркуш	№ докум	Підпис	Дата		

2.6. Проектування бази даних

Для зберігання великої кількості про товари та історію їх продажів було спроектовано базу даних, яка дозволить оперувати усією інформацією. Всі таблиці перебувають у певних відношеннях. Було застосовано два ключові відношення.

Перше це відношення «один до один» воно було використане, коли запис в одній таблиці має тільки один запис у іншій таблиці. Цей взаємозв'язок зустрічається досить рідко, розмістивши всі дані в одній таблиці можна досягти більшої продуктивності, проте він може застосовуватися для розділення логіки. Коли створюється відношення даного типу і поля, що використовуються також унікальні – відношення даного типу може бути створене [11].

Окрім цього також було використано відношення «один до багатьох», цей тип відношень зустрічається найчастіше. Дане відношення необхідне, коли, запис у одній таблиці є частиною запису одного або декількох у іншій таблиці, проте кожен окремо вибраний запис у іншій таблиці стосується одного і тільки одного запису в першій таблиці. Наприклад, відносини між таблицею "Клас" та таблицею "Учень" - адже кожен клас може мати декілька учнів, але кожне учень належить лише до одного класу.

Відношення «багато до багатьох» виникає, коли кожен запис у таблиці

1 стосується декількох записів у таблиці 2, а кожен запис у таблиці 2 стосується декількох записів у таблиці 1. Наприклад, зв'язок між таблицею

«Товар» та таблицею «Постачальник» є багато до багатьох, тому що в одного постачальника може бути багато товарів для поставки, і кожен товар в свою чергу може мати багато постачальників замовленнях. Зазвичай таке відношення реалізується через третю таблицю [3].

В даній розробці воно теж було реалізоване. Воно використовується для створення та зберігання історії продажів кожного продавця. Оскільки кожен продавець може продати безліч товарів і також кожен товар може продати багато

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						46
Зм.	Аркуш	№ докум	Підпис	Дата		

продавців.

Для початку розглянемо сутності «activities» та «users» - рисунок 2.9. Ці сутності описують ключові дані пов'язані з збереженням даних користувачів та їх активності. Сутність users описую дані пов'язані з самим користувачем. Сюди входить унікальний ідентифікатор idusers, який виступає первинним ключем таблиці. А users_type це тип користувача, в даній системі користувач може бути двох типів, якщо 1, то це адміністратор, якщо 2 – касир. Інформація про ім'я самого користувача зберігається в графі users_name. В permission_status описується права доступу касира. Якщо 1 то касир має право лише пробивати чеки, якщо 2 – можливість додавати нові товари в систему.

Дані про дату реєстрації користувача зберігається в reg_data. В свою чергу сутність activities складається з ідентифікатор idactivities, який визначає унікальність даного запису. Дана сутність створена для збереження інформації про те, коли користувачі заходили в систему, тому для визначення який саме користувач заходив в систему було додано вторинний ключ, а саме id_users. Він посилається на сутність users, а точніше на поле idusers. Завдяки цьому ці дві таблиці можна зв'язати між собою. Поле start_time містить в собі інформацію пов'язану з датою та часом початку роботи користувача.

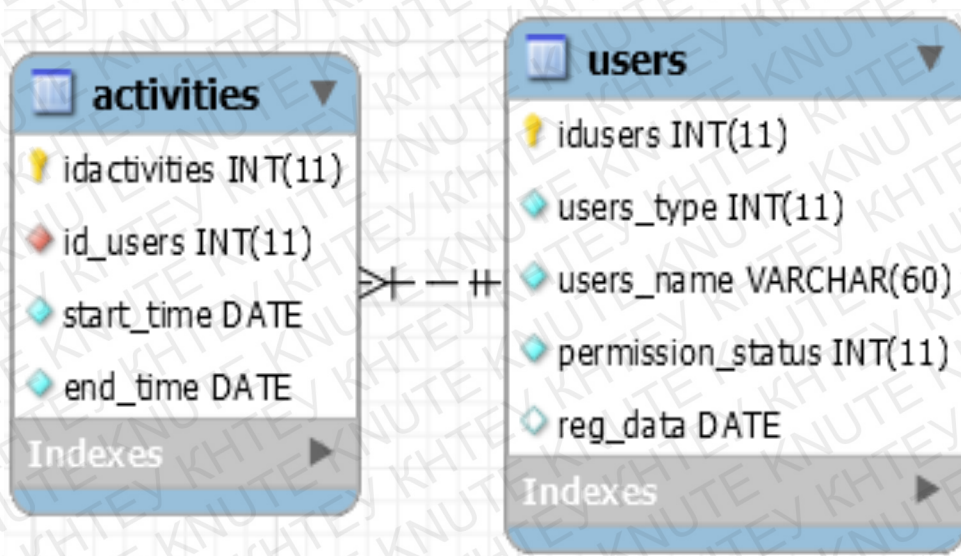


Рис. 2.9 Таблиця сутностей «activities» та «users»

А в `end_time` міститься інформація про те, коли користувач завершив роботу з системою. Як можна помітити, ключова інформація про користувача зберігається в сутності `users`, в свою чергу інформація про активність цього користувача зберігається в сутності `activities` [3].

Відношення між ними встановлене як один до багатьох. Рахується, що один користувач може мати багато записів про активність, проте в одній активності інформація може зберігатися тільки про одного користувача [7].

Також необхідна сутність для збереження даних про товар в системі. Нею виступає таблиця під назвою `items`. В цій таблиці зберігається уся інформація пов'язана з товар мінімаркету. Первинним ключем в цій таблиці виступає `iditems`. Для збереження назви про сам товар передбачене поле під назвою `items_name`. В `incomeDay` зберігається інформація про дату коли товар було додано в систему. Графа `price` відповідає за збереження інформації пов'язаної з ціною товару. Також, кожен товар має свого поставника, в полі `id_supplier` зберігається зовнішній ключ на таблицю з інформацією про поставника товарів. Ця інформація зберігається в таблиці `supplier`. Вона складається з первинного ключа `idsupplier` та поля `name`, в якому власне і зберігається інформація про назву поставника. Між цими сутностями встановлений зв'язок один до багатьох. Це пов'язано з тим, що один поставник може надавати багато товарів, але в товару може бути лише один один поставник. Якщо виникає колізія, при якій один товар можуть надавати двоє різних поставників, товар необхідно ще раз додати

					КНТЕУ 121 02з-15.МР	Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата		

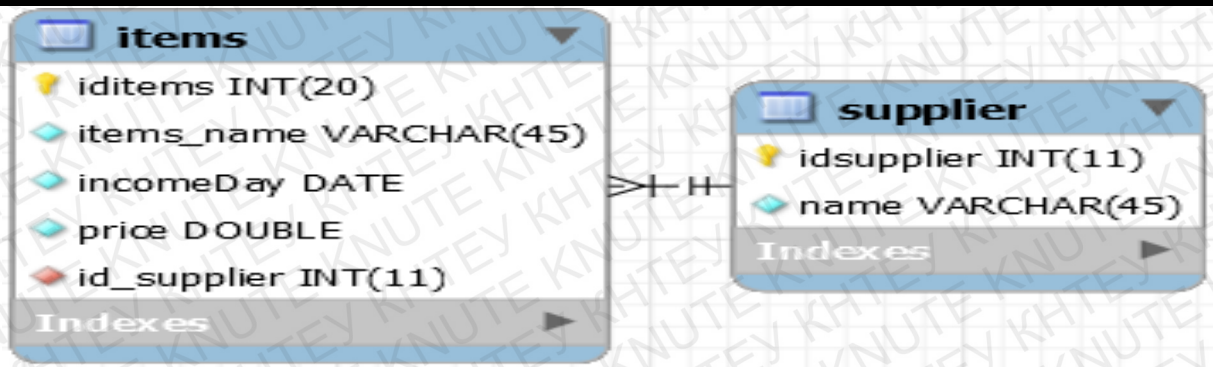


Рис. 2.10 Таблица сущностей «items» та «supplier»

Вже встановлено, що в системі потрібно зберігати інформація про користувачів та про товари.

Однак окрім цього, для ведення звітності необхідно забезпечити зберігання даних про історію продажів.

Саме це забезпечує сутність history, в якій зберігається інформація про продані товари. Первинним ключем виступає поле idhistory. В полі list зберігається весь список покупок. Для збереження інформації про вартість чек використовується поле amount.

Для збереження даних про дату закриття чеку використовується поле date. Також, якщо покупку робить клієнт, який має карту для знижок, відсоток знижки записується у поле discount_percent. Для того, щоб мати статистику про те, який скільки чеків закрит касир, необхідно зберігати інформацію також і про касира, в полі id_seller зберігається зовнішній ключ з посиланням на продавця, який закрит цей чек.

Окрім цього в полі id_card_discount зберігається зовнішній ключ таблиці discount_card, в якій зберігається детальна інформація про карту знижки, якщо така була задіяна. В сутності discount_card роль первинного ключа виконує поле iddiscount_card, а поле short_id зберігає запасний та короткий код по якому можна знайти додатково знайти картку. За допомогою discount_percent можна дізнатися яку знижку надає дана картка, а поле amount зберігає в собі інформацію про загальну суму на яку було куплено в магазині за допомогою цієї картки на знижки.

Дані сутності знаходяться у відношенні оди до багатьох, адже для багатьох покупок може бути лише одна карта.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата		

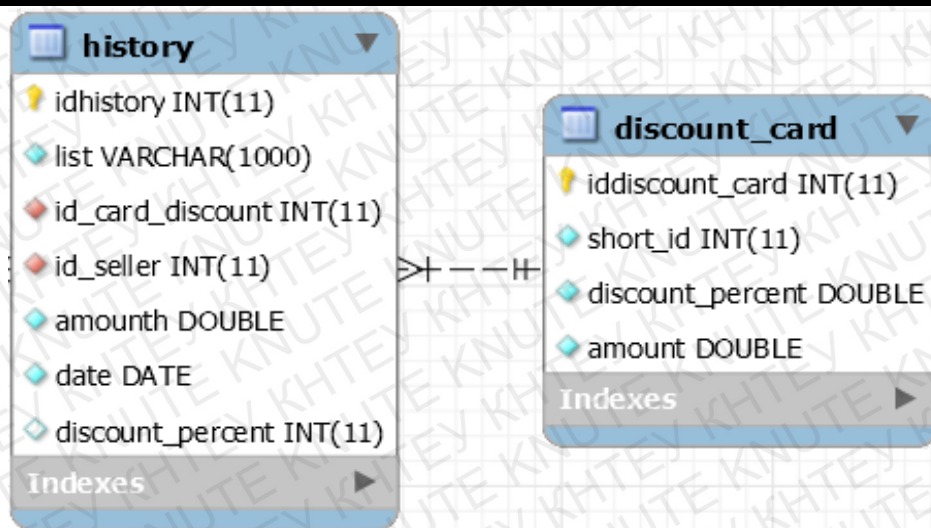


Рис. 2.11 Таблиця сутностей «history» та «discount_card»

На рисунку 2.12 представлена загальна схема сутностей та їх зв'язки. Вище вже було описано загальні сутності та їх ролі. Залишилося ще розглянути відношення багато до багатьох у таких сутностей як history та items. Таке відношення було створене у зв'язку з необхідністю збереження історії покупок для ведення звітності у системі. Це відношення реалізується за допомогою додаткової сутності history_has_items. В ній присутні два поля, а саме history_idhistory та items_iditems. Це означає, що в одній історії може міститися інформація про багато товарів. І в свою чергу, багато товарів можуть входити в одну звітність.

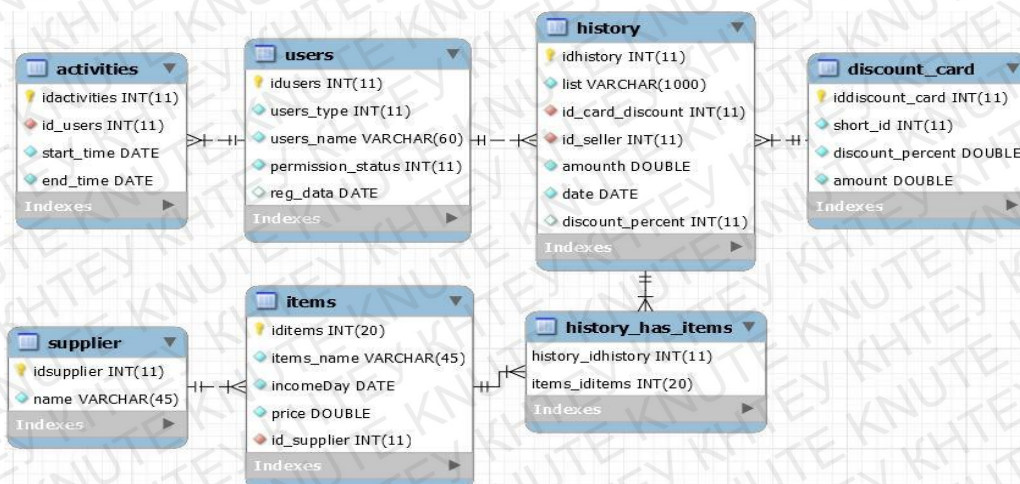


Рис. 2.12 Загальна схема сутностей та їх зв'язків

Для пришвидшення виконання певних елементів програми, було прийнято рішення помістити частину логіки на сторону сервера. На сервері було розміщено вбудовані процедури, тригери та представлення.

Вбудовані процедури мають мету пришвидшити та полегшити виконання певних алгоритмів пов'язаних з маніпуляціями над даними з бази даних, тому вони виконують операції, які користувач програми, в теорії, буде використовувати досить часто. Тригери мають мету забезпечити цілісність даних, які записуються у базу даних та, при необхідності, певним чином редагують записи при додаванні їх у базу даних. Представлення мають на меті полегшити вибірку даних які досить часто вибираються, та скоротити досить громісткий синтаксис при об'єднанні таблиць [11].

2.7. Висновки до Розділу 2

Підсумовуючи весь матеріал можна дійти висновку, що програмна модернізація програмних систем в магазинах є необхідною для сучасного бізнесу та комерції. Особливу увагу потрібно приділити вирішенню проблем описаних вище. Проте перед цим потрібно також модернізувати і наявні системи, щоб вони відповідали всім вимогам, були зручними і зрозумілими у використанні та встановленні пересічному користувачеві персонального комп'ютерів та мобільних пристроїв.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						52
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МІНІМАРКЕТУ

3.1. Розробка та тестування головних функцій системи

Для тестування буде використовуватись скомпільований .exe файл. Тестування програми проводилося на операційній системі Windows 10. Програма тестувалася на коректне відображення компонентів та правильну роботу ключових елементів.

Після запуску програми відкривається початкове діалогове вікно для авторизації у систему (рис. 3.1).

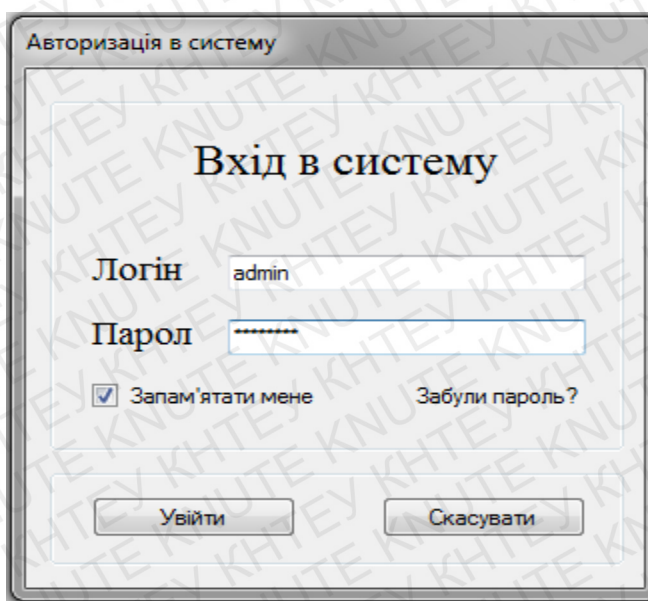


Рис. 3.1 Авторизація в систему

Дана форма складається з двох ключових полів, одне для вводу логіна, інше для пароля. Внизу форми знаходиться дві кнопки: «Увійти» та

					<i>КНТЕУ 121 023-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		21.06.21	Проектування інформаційно- управляючої системи мінімаркету	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		21.06.21		РЗ	53	63
Гарант		Токар В.В.		21.06.21	Реалізація та тестування інформаційно-управляючої системи мінімаркету	Факультет інформаційних технологій 23 курс, 2м група		
Розробив		Тищенко Д.О.		21.06.21				

Скасувати».

Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		21.06.21	<i>КНТЕУ 121 02з-15.МР</i>			
Керівник		Криворучко О.В.		21.06.21				
Гарант		Токар В.В.		21.06.21	<i>Проектування інформаційно-управляючої системи мінімаркету</i>	Стадія	Аркуш	Аркушів
Розробив		Тищенко Д.О.		21.06.21		РЗ	54	63
					<i>Реалізація та тестування інформаційно-управляючої системи мінімаркету</i>	<i>Факультет інформаційних технологій 23 курс, 2м група</i>		

Після того як дані в поля були введені, користувач може натиснути «Увійти», якщо дані ведено коректно, система пропустить далі і користувач почне працювати з іншим вікном. Якщо дані були введені некоректні, з'явиться діалогове вікно, яке проінформує про це користувача. Також присутня функція, яка дозволяє запам'ятати для того, щоб не доводилося повторно вводити логін та пароль. Ця функція діє до перезавантаження комп'ютера.

При натиску на кнопку «Скасувати», закривається діалогове вікно та система завершує свою роботу. При наведенні курсором на напис «Забули пароль» з'являється підказка, яка допоможе відновити доступ до системи.

Саме за цим вікном користувач проводить найбільше часу. Умовно його можна розділити на три частини: верхню, центральну та нижню. У верхній частині знаходиться панель керування під назвою «menuStrip». Це один з компонентів фреймворку Windows Form. В ній розміщені дві кнопки

«Каса» та «Довідка». За допомогою першої кнопки користувач має змогу додати новий товар в систему, за умови відповідних прав доступу, за допомогою другою – отримати довідкову інформацію. Під цим компонентом знаходиться інформація про касира який зараз працює, та про номер чека з яким даний касир зараз працює. В центральній частині програми знаходиться таблиця, яка відображає інформацію про товар, який відсканував касир. Даний компонент має назву «listview». ListView - це прекрасний спосіб відображення інформації та даних файлової системи з XML-файлу чи бази даних. Елемент керування ListView зазвичай використовується для відображення графічного значка, що представляє елемент, а також тексту елемента [2]. Крім того, ListView може використовуватися для відображення додаткової інформації про елемент у підпункті. Наприклад, якщо в елементі управління ListView відображається список файлів, присутня можливість налаштувати елемент управління ListView таким чином, щоб відображати деталі, такі як розмір файлу та атрибути як підпункти. Щоб відобразити інформацію про підрозділ в елементі управління

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						55
Зм.	Аркуш	№ докум	Підпис	Дата		

ListView, потрібно встановити властивість View на View.Details. Крім того, повинні бути створені об'єкти ColumnHeader і призначені їх властивості управління. Після встановлення цих властивостей елементи відображаються у форматі рядків та стовпців, подібних до елемента керування DataGrid. Можливість відображення елементів таким чином робить ListView швидким та простим рішенням для відображення даних із будь-якого типу джерел даних.

При необхідності дані можна відсортувати. Сортування для елемента ListView здійснюється за допомогою метода сортування.

Це дає змогу визначити тип сортування, який слід застосувати до елементів. Це відмінна функція, якщо сортувати лише за елементами. Якщо сортувати за підпунктами, повинні використовувати спеціальні функції сортування елемента [8].

З його допомогою можна створювати таблиці необхідних форм та розмірів. Таблиця складається з 6 стовпців, які відображають ключову інформацію про куплений товар. Після успішної авторизації в систему, касиру відкривається наступне вікно для роботи з системою – рисунок 3.2.



Рис. 3.2 Головне вікно програми для касира

						Аркуш
						56
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 02з-15.МР	

Штрих-код відображає інформацію пов'язану з відсканованим кодом.

«Назва товару» - поле в якому міститься інформацію про назву купленого товару. В «Виробник» - інформація про поставника або виробника даної продукції. Наступні три колонки містять інформацію про кількість та вартість купленої продукції. Також встановлене обмеження на кількість даних в таблиці в розмірі 1000. В середній чек входить близько 25 товарів, для того, щоб уникнути непередбачених помилок було встановлене це обмеження. Для завантаження даних в таблицю необхідно встановити зв'язок з базою даних, в якій зберігається вся інформація [9].

На рис.3.3.продемонстровано метод openConnect, який відповідає за підключення програми до бази даних. Також цей метод використовується при верифікації користувачів.

```
public static void openConnect()
{
    myConnectionString =
    "server=localhost;uid=root;pwd=1234;database=store;"; //підключення до
сервера
    try
    {
        connect = new MySql.Data.MySqlClient.MySqlConnection();
        connect.ConnectionString = myConnectionString;
        connect.Open();
    }
    catch (MySql.Data.MySqlClient.MySqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Рис.3.3. Лістинг «реалізація методу openConnect»

Після того як встановлено з'єднання з базою даних, кожен раз коли користувач сканує товар, система отримує код – унікальний ідентифікатор товару. Після чого робиться запит в базу даних і цьому коду вся інформація про товар. Після чого вона поміщається в компонент таблиці listView. Якщо декілька одиниць одного і того самого товару необхідно просканувати, касир має два

						Аркуш
					КНТЕУ 121 02з-15.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		57

варіанта. Перший – провести кожен товар через сканер, тоді число в колонці «Кількість» збільшується на 1 після кожного сканування.

Або інший варіант – вручну змінити значення в полі «Кількість».

Система автоматично вираховує суму цього товару.

Ця дія виконується кожен раз після сканування штрих-коду. Загальна сума вираховується на основі даних заповнених в таблиці.

В нижній частині вікна знаходяться чотири кнопки управління.

«Очистити чек» - видаляє всі дані з чеку, якщо виникає така необхідність.

«Знижка» дозволяє відкрити діалогове вікно, в якому є можливість ввести код для пошуку карти на знижки або просканувати картку. Якщо це вдалося, тоді вираховується нова сума з врахуванням знижки і відображається знизу в правому куті. В відповідному полі для знижки вказується її відсоток [10].

Заповнення таблиці даними з бази даних описано в лістингу на рисунку 3.4.

```
MySQLCommand command = new MySQLCommand(query, connect); //виконання запиту
using (MySQLDataReader reader = command.ExecuteReader()){
    while (reader.Read()){
        string[] row = { reader.GetInt32(0).ToString(),
reader.GetString(1), reader.GetString(2),
        reader.GetString(3), reader.GetString(4),
reader.GetString(5),reader.GetString(6)};
        var listViewItem = new ListViewItem(row);
        bookList.Items.Add(listViewItem);
    }
}
```

Рис.3.4. Лістинг «заповнення таблиці даними»

Також присутня кнопка «Оплатити». Якщо в чеку немає жодного товару, вона є неактивною. Якщо додається принаймні один товар – активується. При натиску цієї кнопки відкривається діалогове вікно з формою для оплати. Сума яка передається в це вікно вже включає в себе знижку. Користувач може вписати суму, яку йому дав покупець і система автоматично вираховує здачу після натиску на кнопку «Розрахувати». Після цього стає доступно кнопка «Оплатити». Якщо

					КНТЕУ 121 02з-15.МР	Аркуш
						58
Зм.	Аркуш	№ докум	Підпис	Дата		

користувач її натискає, то вважається, що операція завершена і вносяться зміни в БД. За замовчуванням вважається, що покупець розраховується готівкою, якщо ж це не так, касиру просто потрібно поставити галочку на «Безготівковий». Сценарієм використання передбачається, що користувач після цього надасть термінал для безготівкового розрахунку, а квитанції про операцію збереже в касі [7]. На рисунку 3.5 представлена дана форма.

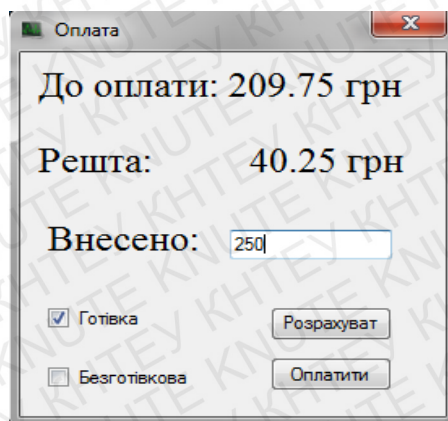


Рис. 3.5. Вікно оплати

Користувач з відповідними правами доступу має можливість додавати новий товар в систему. Для цього, на головній форма, на верхній панелі, потрібно натиснути «Каса», а далі з списку вибрати «Додати товар». Якщо у користувача немає прав доступу, ця функція буде недоступною, якщо є відкриється діалогове вікно для введення інформації про новий товар – рисунок 3.6.

					КНТЕУ 121 02з-15.МР	Аркуш
						59
Зм.	Аркуш	№ докум	Підпис	Дата		

Додати товар

Введіть дані про товар

Штрих-код: 000364

Назва товару: Іграшка "Саморьодько"

Виробник: ТОВ "Іграшкороб"

Ціна: 79.99

Дата: 07.12.2019

Додати Скасувати

Рис. 3.6. Форма додавання товару

Як можна побачити, форма містить три поля для додавання товару, це:

- Штрих-код;
- Назва товару;
- Виробник;
- Ціна;
- Дата.

Це ключові дані які описують товар. Користувач може ввести дані про штрих-код в ручну, або просто просканувати штрих-код потрібного товару, при відкритій формі додавання товару, це код автоматично внесеться в систему. У разі, якщо такий штрих-код уже існує, система видає діалогове вікно з попередженням. Окрім цього, в полі «Ціна» можуть бути введені лише цифрові значення, система не дозволить ввести інші символи. Це саме стосується і графі «Дата» [1]. Сюди можуть бути записані дані лише такого типу, як на рисунку.

3.2. Тестування системи та інших функцій

Також варто проаналізувати скільки пам'яті займає дана програма. Використовуючи стандартні засоби для діагностики Visual Studio на операційній системі Windows 7 було встановлено, що запущена програма використовує орієнтовно 23мб (рисунок пам'яті в режимі очікування, а під навантаженням

						Аркуш
					<i>КНТЕУ 121 02з-15.МР</i>	60
Зм.	Аркуш	№ докум	Підпис	Дата		

близько 100мб в залежності від кількості відкритих вікон та виконуваної роботи. Даний результат повністю задовольняє всі поставлені до системи вимоги. Адже використання пам'яті є досить низьким, що дозволяє програму запускати на ПК з не надто потужним апаратним забезпеченням [5].

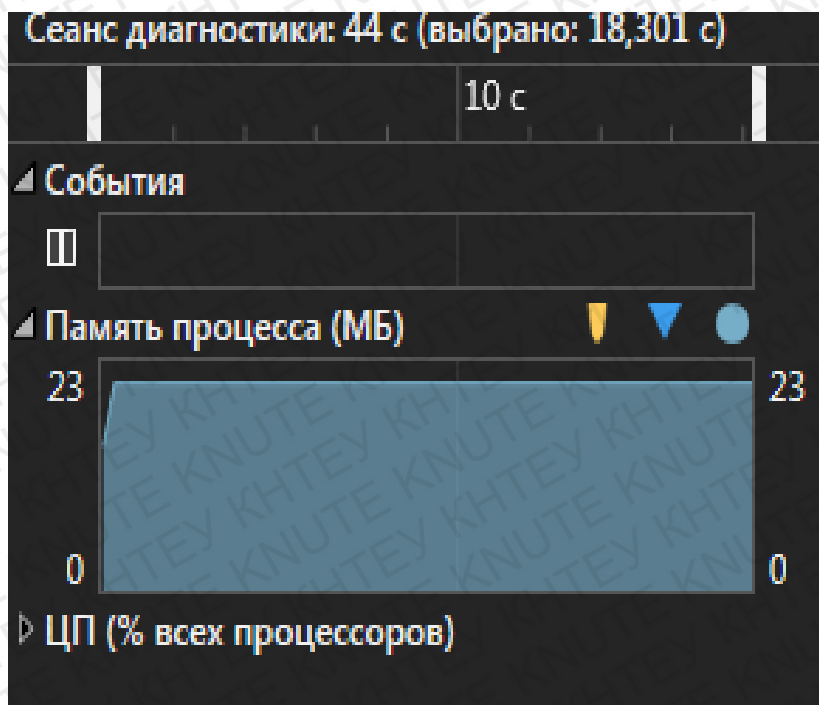


Рис. 3.7. Тестування системи засобами Visual Studio

Також була виконана перевірка за допомогою перегляду навантаження системи в диспетчері завдань – рисунок 3.7. Як видно, при відкритому головному вікні програми та встановленому з'єднанні з базою даних програма використовує не більше 15мб. Проте варто враховувати, що дані з БД можуть використовуватися безпосередньо на локальному сервері, який встановлений разом з додатком ІУС на ПК. Враховуючи, що в базі даних буде не більше 100 000 записів про товари, 1гб оперативної пам'яті, та пам'яті на жорсткому диску повинно вистарчити для комфортної роботи з даною системою. При розрахунку мінімальних системних вимог, також потрібно врахувати затрати, на встановлення стороннього програмного забезпечення для роботи з сканером штрих-кодів [16].

						Аркуш
					<i>КНТЕУ 121 02з-15.МР</i>	61
Зм.	Аркуш	№ докум	Підпис	Дата		

shop.exe #32

00

13 408 K5

shop

Рис. 3.8. Дані з диспетчера завдань

3.3. Висновки до Розділу 3

При використанні об'єктно-орієнтовного підходу зменшується кількість працівників, які залучаються у проект, та зменшуються витрати на реалізацію проекту, але для підтримки проекту і його подальшої модернізації.

Отже, програмний продукт може бути впроваджений та мати подальший розвиток, оскільки він є економічно вигідною за всіма основними техніко-економічними показниками.

					КНТЕУ 121 023-15.МР	Аркуш
						62
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Під час виконання даної роботи було реалізовано додаток для управління мінімаркетом для оптимізації та спрощення продажів. Проаналізовано додатки зі схожим функціоналом та застосовано в даному додатку і компенсовано слабкі сторони. Специфіка даного сегменту полягає у тому, що потрібно забезпечити якісну роботу з базою даних та сканером штрих-кодів. Актуальність тему пов'язана з великою кількістю мінімаркетів, які потребують автоматизації і покращення роботи.

Для розробки даного додатку було застосовано технології C# та .Net. Було використано ключові можливості даних інструментів. Фреймворк windows form дозволив максимально швидко, якісно та зручно розробити графічний інтерфейс система, продемонструвавши всі свої сильні сторони. Було застосовано велику кількість функцій пов'язаних з форматуванням таблиць. Окрім цього було досліджено на застосовано ключові можливості мови програмування C# та інший її функціонал.

Варто також зазначити, що для такого типу програмного забезпечення, надзвичайно зручною виявилася база даних MySQL. Одною з її найбільших переваг можна вважати швидкодію, яка на повністю забезпечували всі поставлені перед нею задачі. Workbench – дозволив якісно та швидко створити ключові представлення для роботою з базою даних.

Широкий функціонал дозволяє за допомогою декількох простих кроків створювати представлення та взаємозв'язки між ними. Також зручною виявилася функція автоматичної побудови діаграм для представлення ключових сутностей системи.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 023-15.МР</i>			
Зав. каф.		Криворучко О.В.		01.11.21	<i>Проектування інформаційно-управляючої системи мінімаркету</i>	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		01.11.21		ВП	63	63
Гарант		Токар В.В.		01.11.21		<i>Факультет інформаційних технологій 23 курс, 2м група</i>		
Розробив		Тищенко Д.О.		01.11.21				
					<i>Висновки та пропозиції</i>			

Діаграми UML дозволяють максимально якісно та доступно продемонструвати ключові можливості системи да її акторів, в зв'язку з чим і були застосовані при проектування системи. Діаграми станів та послідовностей допомогли більше заглибитися в специфіку та глибше в ній розібратися, що дозволили розробити максимально якісний продукт.

Visual studio дозволило пришвидшити процес розробки та спростити певні аспекти розробки. Розроблена програма дозволяє користувачам дізнаватися вартість та іншу важливу інформація про продукт за допомогою штрих-коду. Після дослідження предметної області, було встановлено, що програмою будуть користуватися два типи користувачів, це касир та адміністратор. Саме для цих користувачів і розроблений даний додаток, який спростить процес розробки та дозволить максимально збільшити ефективність їх роботи. Також була розроблена форма авторизація користувачів, яка допомагає максимально спростити процес входу в систему та роботу з нею. Весь функціонал системи було протестовано на швидкодію так коректність роботи. Тести показали, що швидкість роботи програмного забезпечення дозволить максимально якість та швидко виконувати ті функції, які вимагаються від програми. Розроблена система не є ресурсо затратною, що дозволить запустити її навіть на комп'ютерах з слабким апаратним забезпеченням. Мінімальні вимоги передбачають, що версія операційно системи повинна бути принаймні Windows 7.

Завдяки грамотному проектуванні та використання коректних та прагматичних інструментів було створено програмний продукт, який дозволяє спростити роботу з продажами товарів в магазинах і як наслідок оптимізувати його роботу. Використавши аналізи економічної частини розробки програмного продукту вдалося розробити доступний продукт, який знайде своїх покупців на ринку, адже існуючі вже програми не у всіх випадках дозволяють оптимізувати

						Аркуш
					<i>КНТЕУ 121 02м-15.МР</i>	64
Зм.	Аркуш	№ докум	Підпис	Дата		

або автоматизувати роботу для збільшення прибутку та полегшення роботи

						Аркуш
					<i>КНТЕУ 121 02м-15.МР</i>	
Зм.	Аркуш	№ докум	Підпис	Дата		65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація C#. – [Електронний ресурс] - 2019 Режим доступу <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>.
2. Офіційна документація Windows Forms. – [Електронний ресурс] 2019 Режим доступу <https://docs.microsoft.com/en-us/dotnet/framework/winforms/>.
3. Офіційна документація MySQL. – [Електронний ресурс] - 2019 Режим доступу: <https://dev.mysql.com/doc/refman/8.0/en/>.
4. Офіційна документація Workbench – [Електронний ресурс] - 2019 Режим доступу <https://dev.mysql.com/doc/workbench/en/>.
5. Офіційна документація Visual Studio. – [Електронний ресурс] - 2019 Режим доступу <https://docs.microsoft.com/en-us/visualstudio/ide/?view=vs-2019>.
6. Scott W. Amble. The Elements of UML(TM) 2.0 Style / Scott W. Amble.. – 202 с.
7. Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) / Craig Larman.. – 736 с.
8. Michael B. White. Mastering C# (C Sharp Programming): A Step by Step Guide for the Beginner, Intermediate and Advanced User, Including Projects and Exercises / Michael B. White. – Centerville Road Suite 400 Wilmington: Amazon Digital Services LLC. – 517 с.
9. Joseph Albahari. C# 7.0 in a Nutshell: The Definitive Reference 1st Edition / Joseph Albahari. – 41 E University Ave, Champaign, IL 61820,: O'Reilly Media; 1 edition. – 1088 с.
10. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas

					<i>КНТЕУ 121 023-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		27.02.21	<i>Проектування інформаційно-управляючої системи мінімаркету</i>	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		27.02.21		СВД	66	63
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій 2з курс, 2м група</i>		
Розробив		Тишкекно Д.О.		27.02.21				

					<i>КНТЕУ 121 023-15.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		27.02.21	<i>Проектування інформаційно- управляючої системи мінімаркету</i>	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		27.02.21		СВД	67	63
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій 2з курс, 2м група</i>		
Розробив		Тищенко Д.О.		27.02.21				

University Ave, Champaign, IL 61820,: O'Reilly Media; 1 edition. – 616 с.

11. Jamie Chan. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project. / Jamie Chan. – 2711 Centerville Road Suite 400 Wilmington, DE 198: Amazon Digital Services LLC. – 166 с.

12. Don Norman. The Design of Everyday Things: Revised and Expanded Edition / Don Norman.. – 368 с.

13. Офіційна документація фреймворку Iron Barcode. – [Електронний ресурс] - 2019 Режим доступу <https://ironsoftware.com/csharp/barcode/>.

14. Luke Fatooros. 1-PAGE PLANNER: How to Discover your Red-Hot Niche of Cash-Paying Raving Fans. Dominate it. Become the Business Customers Want to Buy From / Luke Fatooros. – 2711 Centerville Road Suite 400 Wilmington, DE 198: Amazon Digital Services LLC. – 84 с.

15. Chris Sells. Windows Forms Programming in C# / Chris Sells. – Boston: Addison-Wesley Professional. – 736 с. – (1st Edition).

16. Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin.. – 432 с. – (1 edition)..

17. Michael Keeling. Design It!: From Programmer to Software Architect / Michael Keeling.. – 360 с.

18. Тищенко Д. Інформаційно-управляючі системи у розробці торгівлі / Збірник наукових статей студентів Software Engineering Комп'ютерних систем. КНТЕУ: Київ, 2021 - с. 169-175.

					<i>КНТЕУ 121 023-15.МР</i>	Аркуш
						68
Зм.	Аркуш	№ докум	Підпис	Дата		

ТЕХНІЧНЕ ЗАВДАННЯ

ЗМІСТ

Вступ

1. Підстави для розробки ІУС
2. Призначення розробки ІУС
3. Вимоги до програмного продукту
 - 3.1 Функціональні характеристики
 - 3.2 Склад та параметри технічних засобів
 - 3.3 Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

1. ПІДСТАВИ ДЛЯ РОЗРОБКИ ІУС

Розробка проводиться у відповідності до плану написання випускних кваліфікаційних робіт здобувачів освіти ос «магістр» спеціальності 121 «Інженерія програмного забезпечення» заочної форми навчання КНТЕУ.

Тема ВКП: «Проектування інформаційно-управляючої системи мінімаркету»

2 ПРИЗНАЧЕННЯ РОЗРОБКИ ІУС

- Інформаційно-управляюча система продажами мінімаркету дозволяє максимально спростити та пришвидшити процес обслуговування клієнта, що в

Зм.	Аркуш	№ докум.	Підпис	Дата	КНТЕУ 121 02з-15.МР			
Зав. каф.	Криворучко О.В.			18.10.21	Проектування інформаційно-управляючої системи мінімаркету	Стадія	Аркуш	Аркушів
Керівник	Криворучко О.В.			18.10.21		ТЗ	69	63
Гарант	Токар В.В.			18.10.21		Факультет інформаційних технологій 2з курс, 2м група		
Розробив	Тищенко Д.О.			18.10.21				
					Технічне завдання			

свою чергу сприяє посиленню продуктивності праці та збільшенню прибутку.

					<i>КНТЕУ 121 023-15.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		18.10.21	<i>Проектування інформаційно- управляючої системи мінімаркету</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Криворучко О.В.		18.10.21		ТЗ	70	63
Гарант		Токар В.В.		18.10.21		<i>Факультет інформаційних технологій 2з курс, 2м група</i>		
Розробив		Тищенко Д.О.		18.10.21				

Метою магістерської роботи є розробка інформаційно-управляючої системи мінімеркату. Необхідно проаналізувати предметну область та дослідити програмні системи з аналогічним функціоналом.

В якості ІУС мінімаркетом необхідно розробити систему, яка забезпечить користувачів максимальним функціоналом. Система повинна працювати з сканером штрих-кодів та забезпечувати можливість входу в систему різних користувачів. Окрім цього необхідно забезпечити зручність системи та інтуїтивно зрозумілий інтерфейс.

За результатами виконаної роботи необхідно розробити автоматизовану систему управління мінімаркетом, яка дозволить пришвидшити та спростити роботу персоналу магазину.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Функціональні характеристики

Програмне забезпечення має виконувати наступні дії:

- сканувати товари за допомогою сканера штрих-кодів;
- працювати з дисконтними картами клієнтів;
- підтримувати можливість виходу та входу в систему, запам'ятовувати користувача в системі;
- автоматично вираховувати знижку на базі наданої дисконтної карти;
- додавати, видаляти та редагувати дані про користувачів;
- додавати дані про новий товар;
- формувати звітність за певний проміжок часу;
- підтримувати одночасну роботу з 40 та більше користувачами.

3.2 Склад та параметри технічних засобів

1) ПК або планшетний комп'ютер з 1024 Мб оперативної пам'яті, встановленою системою Windows 7, 8, 8.1, 10, MacOS, Не менше 200 Мб вільного місця на жорсткому диску. Двоядерний процесор з тактовою частотою

						Аркуш
					<i>КНТЕУ 121 02з-15.МР</i>	71
Зм.	Аркуш	№ докум	Підпис	Дата		

від 1.2 GHz і більше.

2) Наявність встановленої бібліотеки .Net.

3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати в операційних системах Windows, різних поколінь, на яких доступне для встановлення бібліотека фреймворку .Net. Розроблювана бібліотека класів повинна бути пристосована до використання у інформаційних системах та програмних засобах. Розробку виконувати з використанням бібліотек та технологій мови C# в середовищі програмування Visual Studio 2017 з використанням .Net.

4 СТАДІЇ РОЗРОБКИ

В ходів реалізації роботи проект повинен пройти крізь наступні стадії розробки:

- аналіз предметної області;
- проектування архітектури;
- реалізація архітектури;
- реалізація графічного інтерфейсу;
- тестування результатів розробки;
- оформлення супровідної документації;
- задача роботи.

5 ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						72
Зм.	Аркуш	№ докум	Підпис	Дата		

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

					<i>КНТЕУ 121 02з-15.МР</i>	Аркуш
						73
Зм.	Аркуш	№ докум	Підпис	Дата		

