

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
УПРАВЛІННЯ МАГАЗИНОМ
SOFTWARE DEVELOPMENT TO MANAGE A STORE

Виконав(ла): студент(ка) 4 курсу, групи КНТ–210СП

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ГУЛАК В.М.

(ПРИЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ТЕРЕЩЕНКО Е.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

« _____ » _____ 2023 року

З А В Д А Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГУЛАКА Владислава Максимовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для управління магазином. Software Development to Manage a Store

керівник проєкту (роботи) к.т.н., доцент ФЕДОРОНЧАК Тетяна Василівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «04» квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 21 травня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

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

1. Аналіз проблем та постановка завдань дослідження. 2 Матеріали і методи
3 Опис програми 4. Експлуатація, тестування та експериментальне
дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1–4 Основна частина	ФЕДОРОНЧАК Т.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю, ст.викл.		

7. Дата видачі завдання «04» квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки	2 тиждень	Розділ 2
4	Розробка архітектури програми	2 тиждень	Розділ 3
5	Розробка програми	3–4 тижні	Розділ 3,4
6	Тестування та експериментальне Дослідження програмного забезпечення	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та Документів до неї	6 тиждень	Додатки
8	Нормоконтроль та рецензування	7 тиждень	
9	Захист роботи	8 тиждень	

Студент(ка)

(підпис)

Владислав ГУЛАК

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Тетяна ФЕДОРОНЧАК

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
92 сторінки, 39 рисунків, 1 таблиця, 3 додатки, 16 джерел.

МАГАЗИН, КОМП'ЮТЕРНА ПРОГРАМА, C#, ЧЕК, ТОВАР, VISUAL
STUDIO, .NET.

Об'єкт дослідження – програмні засоби для управління магазином.

Предмет дослідження – програмний застосунок для управління магазином.

Мета роботи – створення програмного застосунку для управління магазином, контролю за базою даних товарів та обслуговування покупців.

Матеріали та технічні засоби: мови програмування C#, .NET Framework, SQL, робота з даними Microsoft Access, ПК з процесором AMD Athlon II x2, 4 ГБ ОЗУ під управлінням системи MS Windows 7.

Результати. Створено програмний Windows-застосунок для простого та зручного управління магазином, роботи з товарами та персоналом, контролю за продажами.

Висновки. Розроблено автоматизовану програмну систему для полегшеного управління власним магазином.

Галузь використання – сфера торгівлі, адже програмний продукт можна адаптувати для більшості типів магазинів як роздрібної так і для оптової торгівлі.

ABSTRACT

Explanatory note to the bachelor's thesis: 92 pages, 39 pictures, 1 table, 3 appendixes, 16 sources.

STORE, COMPUTER PROGRAM, C#, CHECK, ITEM, VISUAL STUDIO, .NET.

The object of research is software for store management.

The subject of research is a software application for store management.

The purpose of the work is to create a software for store management, control over the product database and customer service.

Materials and technical means: programming languages C#, .NET Framework, SQL, working with Microsoft Access data, a PC with an AMD Athlon II x2 processor, 4 GB of RAM running the MS Windows 7 system.

The results. A Windows software was created for simple and convenient management of the store, work with goods and personnel, sales control.

Conclusions. An automated software system has been developed for easier management of your own store.

The field of use is the field of trade, because the software product can be adapted for most types of stores, both retail and wholesale.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	6
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Опис предметної області	10
1.2 Аналіз аналогів	12
1.2.1 UniproRetail	12
1.2.2 ZStore	13
1.3 Постанова завдання та функціональні вимоги до застосунку.....	15
1.4 Не функціональні вимоги до застосунку	18
1.5 Вимоги до інтерфейсу.....	19
1.6 Висновки за розділом 1	19
2 Матеріали і методи.....	20
2.1 Вибір інструментів розробки	20
2.1.1 Вибір мови програмування	20
2.1.2 Вибір компілятора.....	22
2.1.3 Вибір СУБД	23
2.2 Проєктування бази даних	23
2.3 Висновки за розділом 2	26
3 Опис програми.....	26
3.1 Реалізація функціоналу системи.....	26
3.2 Створення користувацького інтерфейсу.....	31
3.3 Висновки за розділом 3	32
4 Експлуатація, тестування та експериментальне дослідження програми	33
4.1 Умови застосування та призначення програмного продукту.....	33
4.2 Комплект постачання програмного продукту.....	34
4.3 Характеристика програми	34
4.4 Інструкція з експлуатації програми.....	35

	7
4.4.1 Інсталяція програмного застосунку	35
4.4.2 Робота з програмою	35
4.4.2.1 Робота з товарами.....	36
4.4.2.2 Робота з категоріями товарів	39
4.4.2.3 Робота з персоналом	41
4.4.2.4 Оформлення замовлення та створення чека.....	43
4.4.2.5 Архів замовлень	46
4.5 Тестування програми	46
4.6 Висновки за розділом 4	48
Висновки	49
Перелік джерел посилання	50
Додаток А Технічне завдання	52
Додаток Б Текст програми	57
Додаток В Слайди презентації.....	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ОС – Операційна система;
ПЗ – Програмне забезпечення;
SMS – Short Message Service;
API – Application Programming Interface;
MS – Microsoft;
VS – Visual Studio;
SQL – Structured Query Language;
ПК – Персональний комп'ютер;
ТМЦ - Товарно-матеріальні цінності;
ФОП – Фізична Особа Підприємець;
БД – База даних;
СУБД – Система Управління Базами Даних.

ВСТУП

На сьогоднішній день автоматизовані системи керування використовуються у багатьох сферах нашого життя. Кожного дня ми так чи інакше з ними зтикаємося. Хоча і не прямо, але ми стаємо частиною цих систем: влаштовуємось на нову роботу, робимо покупку у магазині, оформлюємо та отримуємо пошту та ін.

У нашій країні наразі наявні проблеми з системами управління. Одна з основних проблем це відсутність товарних чеків, а слідуючи з цього неможливість дотримання закону про права споживачів. Так ця проблема не стосується всіх, але невеликі підприємці частіше всього не використовують товарні чеки. Адже для них потрібні касові апарати, або дороге ПЗ. А з недавнім прийняттям закону про товарні чеки, питання про їх введення починає становитися гострим для малих та деяких середніх підприємців.

Також до проблем можна віднести складний ручний облік товару, його потрібно постійно вписувати при прийомі та поверненні чи списанні. В наших реаліях це тони паперу, тому що працівників важко навчити роботі з перевантаженими системами, і дуже часто, що вже готові рішення потребують великих ресурсів ПК. З цього і з'являється проблема для підприємців та роботодавців, адже у реаліях війни в яких дрібні магазини ледь тримаються на плаву через зростання цін, вкладати у системи управління просто нічого.

Однак при наявності безплатного ПЗ яке не потребує надто потужного ПК, більшість підприємців би перейшли до програмних систем управління магазином.

Розробка та використання системи управління магазином на сьогоднішній день є актуальним питанням особливо для нашої країни.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Опис предметної області

Магазин – підприємство чи заклад роздрібної торгівлі. Складові крамниці: торговельний зал або кілька залів для обслуговування покупців, каса, підсобні приміщення, де приймають, зберігають і готують до продажу товари; адміністративно–побутові й технічні приміщення[1].

Продавець – особа, яка здійснює продаж товарів покупцям, приймає товар, також може виконувати облік товарів у магазині.

Законопроект «Про внутрішню торгівлю» станом на вересень 2011 містить таке визначення: Магазин — це об'єкт торгівлі, розташований в окремій капітальній споруді або приміщеннях іншої капітальної споруди або тимчасової споруди некапітального типу або їхніх частинах з торговельними, підсобними, адміністративно–побутовими приміщеннями, а також приміщеннями для приймання, зберігання, підготовки та продажу товарів, оснащений торговим та інженерним устаткуванням, яке забезпечує здійснення торговельної діяльності [1].

До основних видів крамниць можна віднести:

- роздрібний магазин;
- оптовий магазин;
- інтернет–магазин;
- торговий центр;
- супермаркет;
- склад–магазин.

Найпоширенішими у нашій країні можна назвати роздрібні магазини та супермаркети.

Супермаркет є дуже зручним для використання людиною. Всі товари на виду, приємне оформлення інтер'єру, розслаблююча музика, великий

вибір товарів. Тобто все зроблено для того, щоб покупцю не потрібно було шукати потрібні товари по різних магазинах, щоб все було в одному місці і з приємними цінами і тим самим приносити прибутки для власника.

Ми будемо досліджувати звичайні роздрібні немережеві магазини, адже основна проблематика обраної мною теми стосується саме таких ФОП. Кожен магазин цього сегменту є меншим по асортименту, задля того щоб мінімізувати ризик збитків на непопулярних товарах, але це не говорить про те, що роботи з паперами там менше. Навпаки, якщо брати дрібні магазини то вся робота з обліком товарів (прийомом, списанням) лежить на руках продавців.

Для прикладу роботи був використаний риболовний магазин. Це магазин направлений на продаж необхідного риболовного інвентарю для риболовів як любителів, так і спортсменів.

Основні типи товарів риболовного магазину:

- вудилища;
- приманки;
- спеціалізований інструмент;
- гачки;
- волосінь та шнури;
- дрібні принади(грузила, поплавці, стопори, годівниці та ін.);
- туристичні меблі та спорядження;
- прикормки та насадки.

Такий тип магазину не так поширений як наприклад продуктовий чи інші, але структура роботи у всіх майже однакова.

1.2 Аналіз аналогів

На сьогоднішній день є багато реалізацій систем управління магазином вони розповсюджуються як на платній так і на безоплатній основі.

1.2.1 UiproRetail

UniproRetail – це програма для автоматизації магазинів, мінімаркетів, бутиків [2].

Продукт має дуже широкий функціонал:

- бек–офіс та робоче місце продавця в одній ліцензії;
- модуль друку цінників та етикеток на продукцію;
- відображення фотографій в екрані касира та в бек–офісі;
- підтримка великого асортименту торгового обладнання;
- одночасне ведення обліку декількох організацій;
- робота з різними складами;
- 15 звітів для аналізу роботи;
- відправка звітів на e–mail власника [2].

Застосунок підтримує роботу як з хмарним так і з локальним сховищем і не потребує постійного з'єднання з інтернетом, має систему авторизації , та велику кількість звітів за різними критеріями. Дозволяє виконувати друк цінників завдяки великій бібліотеці шаблонів.

Програма підтримує роботу з будь–якими USB–сканерами штрихкодів, принтерів чеків, принтерами друку штрихкодів, офісними принтерами. Також програма працює на POS–терміналах, підтримує фіскальні реєстратори, електронні ваги та ін. [2].

Серед значних мінусів можна назвати відсутність підтримки української мови.

Вигляд одного з вікон програми представлено на рис. 1.1.

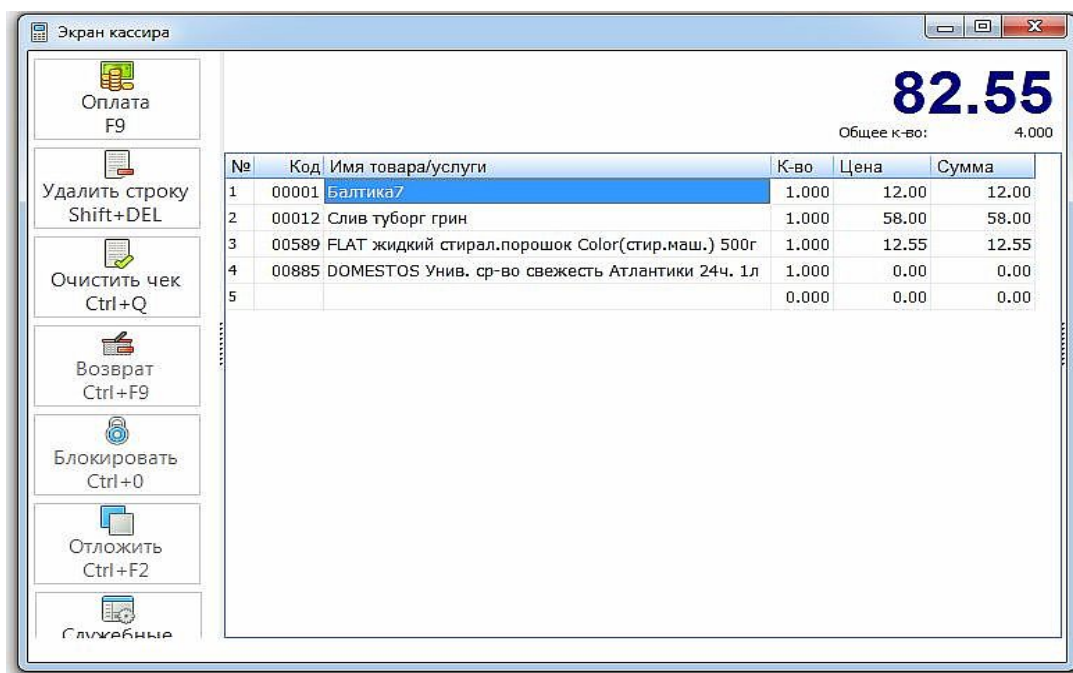


Рисунок 1.1. – Вигляд вікна програми UniproRetail

1.2.2 ZStore

ZStore – безкоштовна облікова програма з відкритим кодом (open-source) та вебінтерфейсом, що дозволяє працювати з програмою з будь якого місця і з будь якого мобільного пристрою.

Програма призначена для використання малим бізнесом зі спрощеною формою обліку, який не використовує повноцінний бухгалтерський облік. Проект виконаний у вигляді вебсайту і може бути розгорнутий на будь якому малобюджетному хостингу, або в локальній мережі [3].

Особливості рішення:

- управління складами, складська логістика;
- закупівля;
- продаж;
- облік платежів та взаєморозрахунки з контрагентами;
- облік по партіях товару та серіях виробника;

- управління користувачами та розділення доступу;
- елементи CRM, робота з лідами;
- можливість локалізації;
- звіти по продажам, закупкам, руху ТМЦ;
- послуги, наряди на виконання робіт;
- списання на виробництво та оприходування готової продукції;
- облік обладнання;
- API для обміну з іншими інформаційними системами;
- підтримка сканера (клавіатурного) штрих коду;
- підтримка принтера чеків та етикеток;
- розділення доступу між філіями (наприклад, торговими точками);
- інтеграція з Opencart та WooCommerce;
- SMS розсилки;
- інтеграція з Новою Поштою;
- інтеграція з CheckBox;
- вбудований (безкоштовний) програмний РРО;
- інтеграція з paperless.com.ua;
- різноманітні модулі (управління проектами, база знань) [3].

Значним плюсом цього рішення це open-source направлення та використання вебінтерфейсу, адже це дозволяє значно розширити доступність для різних клієнтів. Також рішення підтримує українську мову.

Вигляд головної сторінки програмного застосунку ZStore представлений на рис. 1.2.

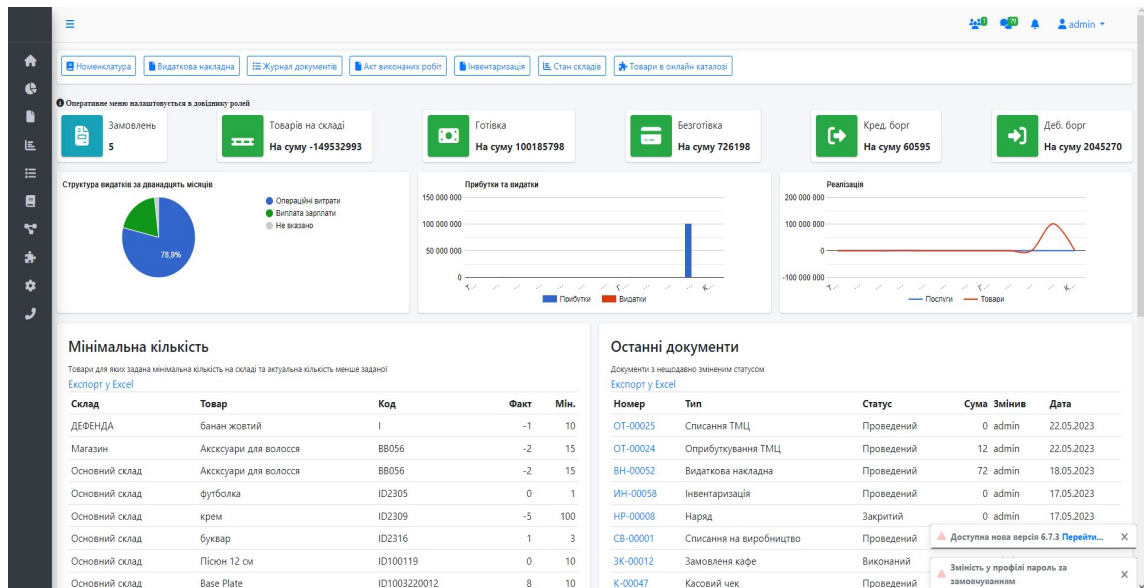


Рисунок 1.2 – Головна сторінка ZStore

Розглянуті системи мають багато можливостей, але дуже складні для навчання, а функціонал може бути занадто перевантаженим для маленьких магазинів.

1.3 Постанова завдання та функціональні вимоги до застосунку

Завданням є створення програмного продукту для управління магазином з дружнім інтерфейсом, орієнтованим на легке навчання роботі з нею.

Для визначення функціональних вимог до програмного продукту ми маємо визначити основні потреби і функціонал для користувача. Для цього ми створимо сценарій роботи користувача за допомогою діаграми прецедентів UML.

UML — уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-

моделлю. UML був створений для визначення, візуалізації, проєктування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація [11].

Діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі [12].

Діаграма прецедентів показує різні варіанти використання та різні типи користувачів системи і часто супроводжується іншими типами діаграм. Варіанти використання представлені колами або еліпсами. Актори (дійові особи) часто зображуються у вигляді паличок [12].

Суть діаграми прецедентів полягає в тому, що проєктована система подається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система під час діалогу з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів із системою [12].

Сценарій представлений діаграмою прецедентів на рис. 1.3.

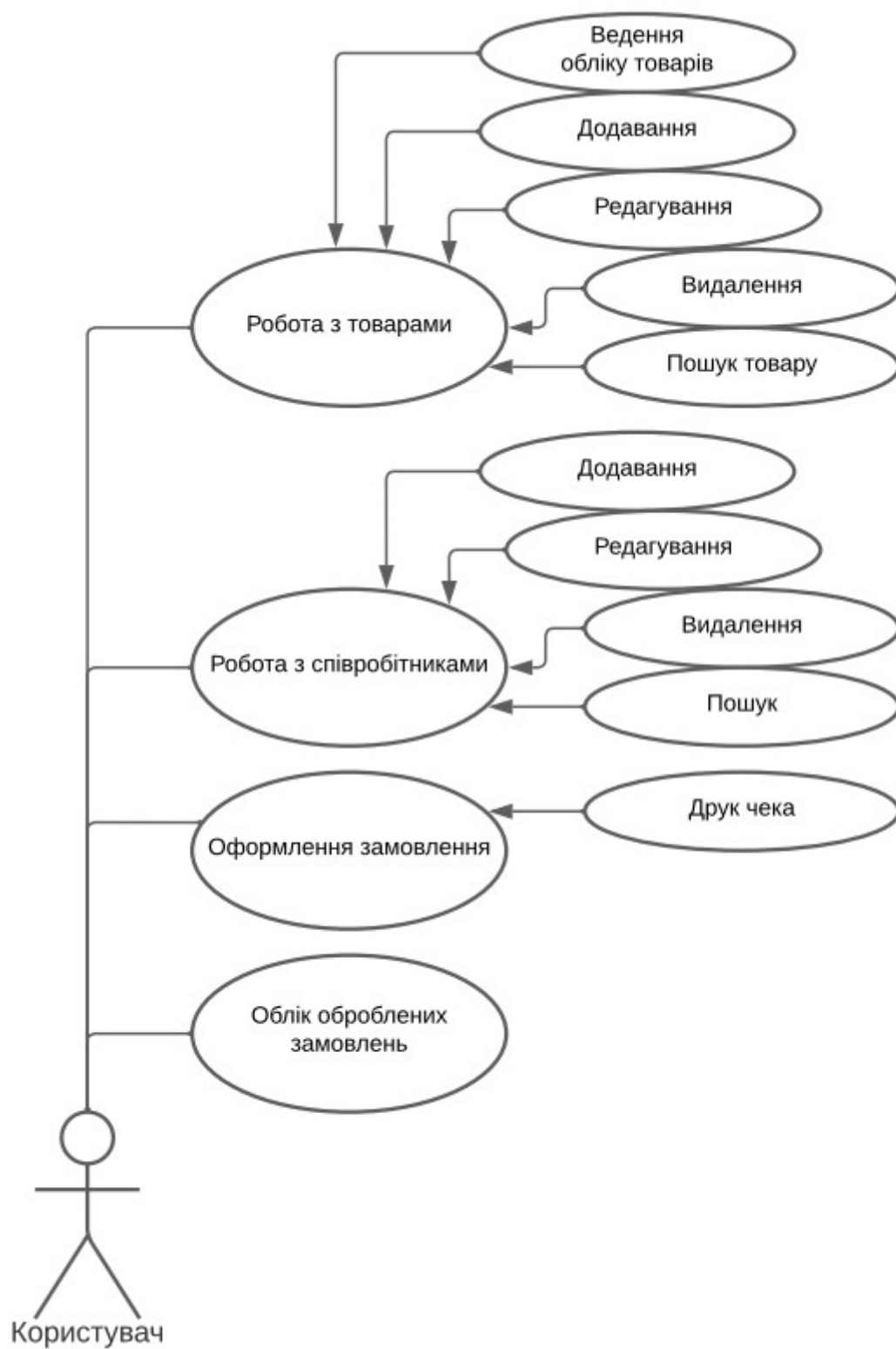


Рисунок 1.3 – Діаграма прецедентів користувача

З огляду на діаграму можна сформувати наступні вимоги для програмного застосунку для управління магазином:

- відображення всіх наявних товарів у магазини;
- додавання нових товарів до бази даних;
- редагування вже існуючих у базі товарів
- видалення товарів з бази даних;
- пошук необхідного товару за різними критеріями;
- ведення залишку товару на складі;
- додавання, редагування, видалення працівників з бази;
- пошук необхідного працівника у базі;
- перегляд інформації по вже обробленим замовленням з указанням дати та суми замовлення;
- оформлення замовлення покупцю;
- друк товарного чека;
- підтримка української мови для використання у сфері послуг;
- швидка робота та невисокі системні вимоги.

1.4 Не функціональні вимоги до застосунку

Для роботи програмного засобу необхідний IBM-сумісний ПЗ на базі процесорів Intel Core 2 Duo або AMD Athlon II x2 з тактовою частотою не менше 2 GHz або вище, об'ємом оперативної пам'яті 2 ГБ та вище, жорсткий диск або твердотільний накопичувач об'ємом від 256 ГБ, SVGA-монітор, маніпулятор типу «миша» та клавіатура. Для виводу на друк потрібен принтер. Операційна система MS Windows 7 та вище зі встановленим розповсюджуваним пакетом Microsoft Report Viewer 2015 Runtime.

1.5 Вимоги до інтерфейсу

Інтерфейс повинен бути інтуїтивно–зрозумілим і простим для простого навчання та роботи. Інформація повинна бути поданою чітко та структуровано.

1.6 Висновки за розділом 1

В цьому розділі був проведений аналіз предметної області та проблеми у сфері торгівлі для малих підприємств. Виконав аналіз існуючих аналогів на ринку. Проаналізував основні потреби для користувача на даний час і на їх основі сформував задачу та функціональні вимоги для створення програмного продукту.

Завдання дипломної кваліфікаційної роботи полягають в наступному:

- аналіз предметної області;
- проєктування програмного забезпечення;
- реалізація програмного забезпечення.

2 МАТЕРІАЛИ І МЕТОДИ

В даному розділі буде проаналізовано та обрано мову програмування для реалізації поставленого завдання та засоби для роботи з даними і програмні комплекси для оформлення пояснювальної записки.

2.1 Вибір інструментів розробки

2.1.1 Вибір мови програмування

Для реалізації програмного застосунку була обрана об'єктно-орієнтована мова програмування C#.

C# – це об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research [4].

В той час, коли такі мови, як Python і PHP, існують досить тривалий час, C# вважається молодого мовою програмування. Розроблена у 2000 році. Сьогодні Андерс Гейлсбер продовжує працювати у компанії Microsoft як провідний архітектор C#. Андерс Гейлсберг також відомий як головний архітектор Delphi й перший автор Turbo Pascal [13].

Базовий синтаксис C# подібний до мов стилю C, таких як C, C++ і Java. Ця мова програмування найбільше відповідає стандарту Common Language Infrastructure (CLI) [13].

C – епохальна мова програмування, що виникла в 70-х. Вона змогла замінити асемблер і стала основою для інших мов. Після Cі вийшов ще C++, що підтримує парадигми об'єктно-орієнтованого та узагальненого програмування. C++ – досі популярна мова програмування [5].

C# — це мова, яка має сувору типізацію. У ній є автоматичний збір сміття — програмісту не потрібно дбати про ручне звільнення пам'яті [5].

До основних відмінностей цих двох мов можна віднести рівень складності та цілі поставлених задач. Код C++ має специфічні показники змінних на відміну від C#. Історично C# тісно пов'язаний з .NET Framework.

Написання коду на C# набагато зручніший ніж C++ завдяки більш зручному синтаксису. На відміну від C++, C# не вимагає глибокого розуміння устрою мови, як це потрібно для C++, завдяки набору готових для вирішення завдань модулів.

.NET Framework — програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Розробка Microsoft технології .NET Framework почалась у 1999 році. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість [6].

Для роботи та взаємодії з базою даних використана діалогова мова програмування SQL. Це мова для створення і здійснення запитів для внесення змін до бази даних. Завдяки ній можна виконувати пошук, додавання і видалення даних з БД [8].

Перша версія SQL була розроблена на початку 1970-х років у IBM. Ця версія мала назву SEQUEL і була призначена для обробки та пошуку даних, що містилися в реляційній базі даних IBM, System R. Мова SQL надалі була стандартизована Американськими Держстандартами (ANSI) в 1986. На початку SQL була запланована як мова запитів і управління даними, а пізніші модифікації SQL створені продавцями системи управління базами даних, які додали процедурні конструкції, control-of-flow команд і темпоральні розширення мов [8].

Переваги SQL:

- незалежність від конкретної системи управління базою даних;
- наявність стандартів;

- декларативність(програміст описує дані які потрібно витягнути або модифікувати) [8].

Недоліки SQL:

- складність;
- відхилення від стандартів;
- складність роботи з ієрархічними структурами [8].

2.1.2 Вибір компілятора

Вибір компілятора пав на MS Visual Studio який є титульним для мови C#. Це серія продуктів компанії Microsoft з інтегрованим середовищем розробки ПЗ та багато інструментальних засобів для роботи [7].

Visual studio включає у себе наступні компоненти:

- Visual C++;
- Visual C#;
- Visual Studio Debugger [7].

Він дозволяє розробляти ПЗ з графічним інтерфейсом і підтримує технологію Windows Forms.

Основною перевагою також можна назвати Велику кількість сторонніх додатків у мережі інтернет пошук яких інтегрований у саму Visual Studio.

Існують також інші компілятори для C#. Наприклад SharpDevelop, який повністю написаний на C#, і є альтернативою до Visual Studio, зі схожим з попередніми версіями інтерфейсом, але в нього завершилася підтримка розробниками з 2016 року і проєкт далі не розвивається, підтримує .NET Framework до версії 4.5.1 [4].

2.1.3 Вибір СУБД

Система управління базами даних— набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних [14].

Основні характеристики СУБД:

- контроль за надлишковістю даних;
- несуперечливість даних;
- підтримка цілісності бази даних (коректність та несуперечливість);
- цілісність описується за допомогою обмежень;
- незалежність прикладних програм від даних;
- спільне використання даних;
- підвищений рівень безпеки [14].

Для роботи з базою даних була обрана СУБД Microsoft Office Access. Access – це система управління базами даних, програма, що входить до складу пакету офісних програм Microsoft Office. Має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних [9].

Включає до себе такі елементи як конструктор таблиць, конструктор форм, конструктор запитів мовою SQL, конструктор звітів, які можна вивести на друк [9].

2.2. Проєктування бази даних

Схема даних складається з 5 сутностей і 4 зв'язків між ними.

Сутність «Товари» представляє собою список усіх товарів які мають на складі. Має наступні поля: КодТовару (ключ), Назва (назва товару), КодКатегорії (посилання на сутність Категорії яка відображає належність до

відповідної категорії товарів), ЦінаТовару (ціна товару), Виробник (виробник товару), ЗалишокНаСкладі (кількість наявного товару).

Сутність «Категорії» представляє собою список категорій товарів. Має наступні поля: КодКатегорії (ключ), НазваКатегорії (назва категорії товару).

Сутність «Персонал» зберігає інформацію про працівників. Має наступні поля: КодПерсоналу (ключ), ПІБ. (прізвище, ім'я та по батькові працівника), Посада (посада робітника), Телефон (зберігає контактний телефон працівника).

Сутність «Чек» представляє собою проміжну таблицю для формування чеків, зберігає у собі інформацію про продаж для перенесення у сутність «Продажа». Має наступні поля: КодЧека (ключ), Дата (дата продажу), КодТовару (інформація про код товару), Назва (назва товару), ЦінаТовару (ціна товару), Кількість (кількість проданого товару), КодПерсоналу (інформація про робітника), ЦінаПродажу (сума товарів одного типу в залежності від кількості).

Сутність «Продажа» зберігає інформацію про всі здійснені продажі для контролю за проданими товарами. Має наступні поля: КодПродажу (ключ), НомерЧека (показує якому чеку відповідає продаж), Дата (дата продажу), КодТовару (унікальний ідентифікатор товару), Назва (назва товару), ЦінаТовару (ціна товару), Кількість (кількість проданого товару), КодПерсоналу (унікальний ідентифікатор робітника), ЦінаПродажу (сума товарів одного типу в залежності від кількості).

Схема БД представлена на рис. 2.1.

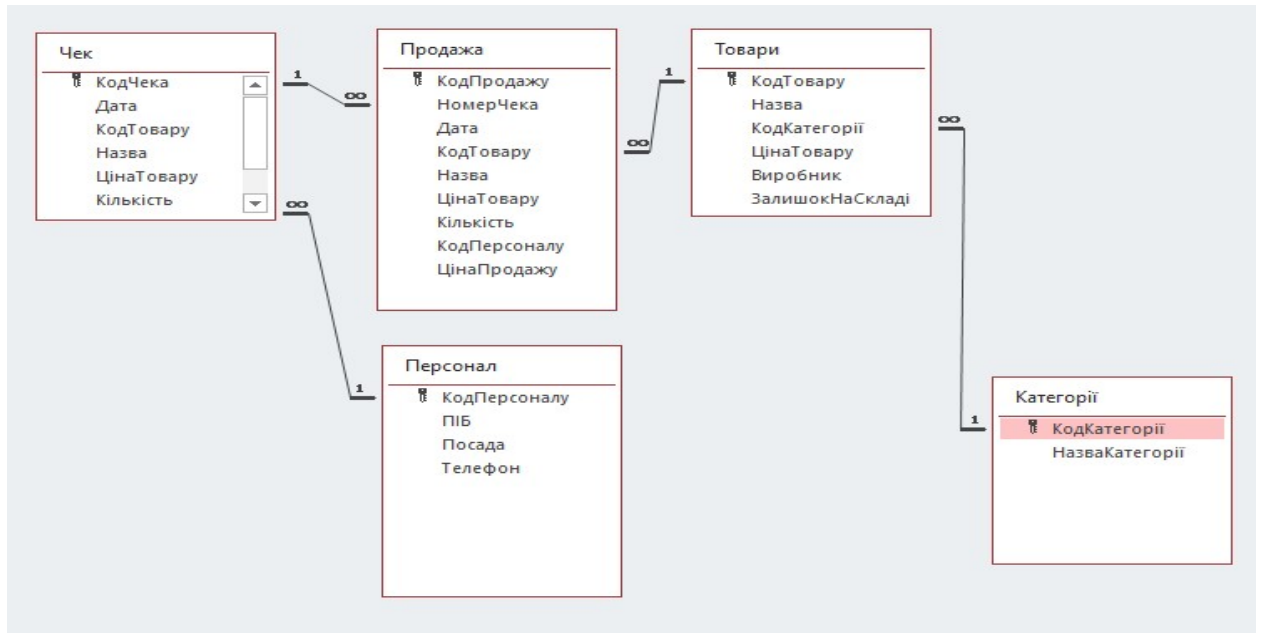


Рисунок 2.1 – Схема даних

Зв'язок між таблицями Категорії і Товари один–до–багатьох, бо одна категорія може відноситись до багатьох товарів.

Зв'язок між таблицями Товари і Продажа один–до–багатьох, тому що один товар можна продати багато разів.

Зв'язок між таблицями Чек і Продажа один–до–багатьох, бо за одним чеком можна продати багато товарів.

Зв'язок між таблицями Персонал і Чек один–до–багатьох, бо один робітник може зробити багато чеків.

2.3 Висновки за розділом 2

У розділі 2 було проведене дослідження технічної частини розробки. Для розробки застосунку було обрано мову програмування C# в середовищі розробки Microsoft Visual Studio. Робота з БД обрано СУБД MS Access.

Проведений аналіз та визначена детальна структура бази даних для роботи програмного забезпечення.

Розроблено базу даних на основі проведених досліджень.

3 ОПИС ПРОГРАМИ

3.1. Реалізація функціоналу системи

До початку роботи над реалізацією програмного продукту треба сформувати, як буде функціонувати система на основі вже сформованих функціональних вимог. Для полегшення подальшої розробки було сформовано візуальне представлення функціоналу системи, який зображений на рис. 3.1.

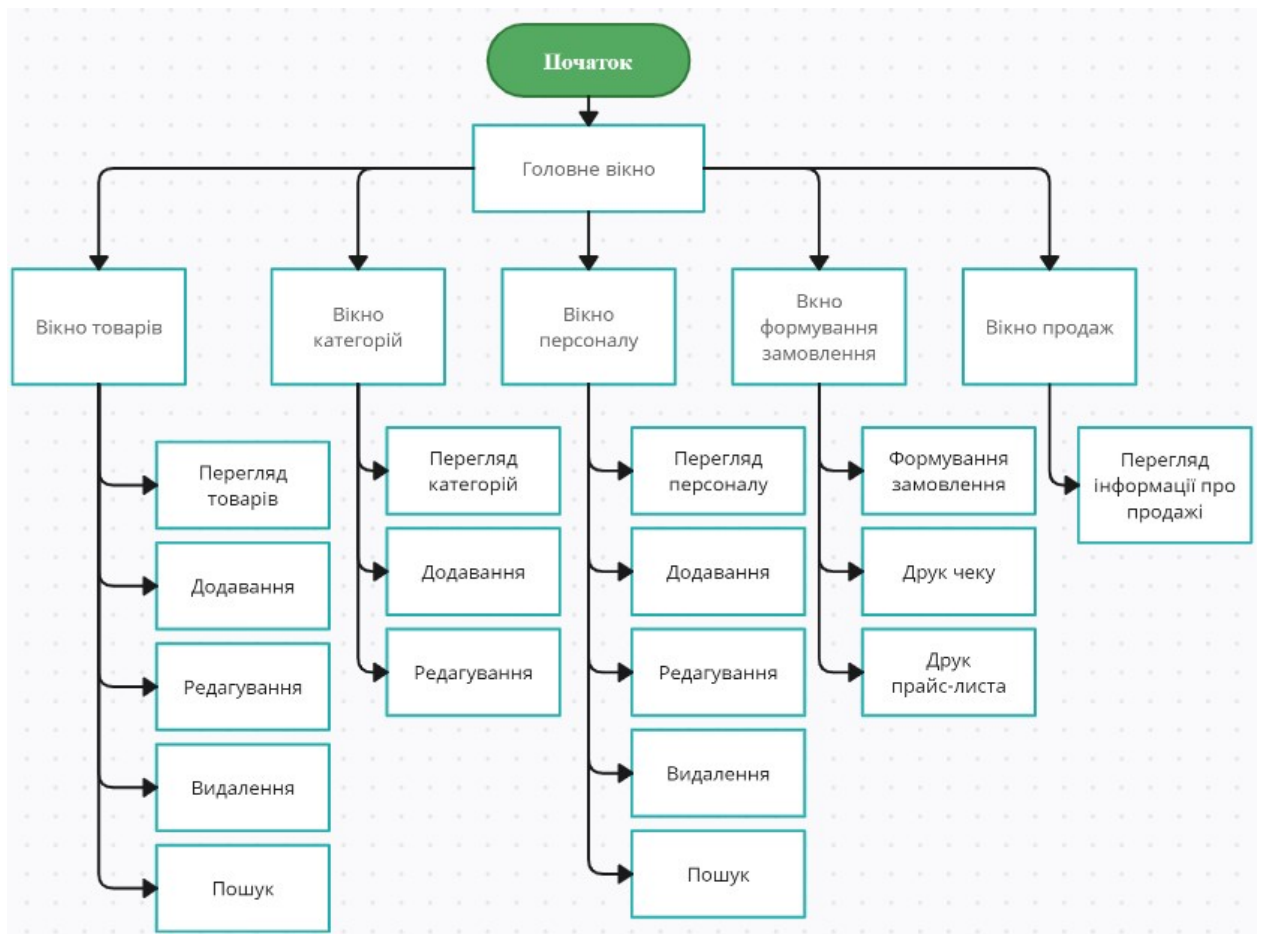


Рисунок 3.1 – Функціонал системи

Для реалізації потрібного функціоналу було створено 8 користувацьких форм:

- MainPage;
- Goods;
- Categories;
- Personel;
- OrderingHistory;
- Ordering;
- Check;
- PriceList.

Форма MainPage використовується для навігації програмою, тобто виконує функцію хаба для переходу на потрібну форму.

Форма Goods використовується для перегляду всіх наявних у магазині товарів, контролю залишків товару на складі. Дозволяє виконувати додавання, редагування та видалення товарів зі списку, для реалізації процедур приходу товару та його списання. Форма дозволяє виконувати пошук товарів за різними критеріями.

Форма Categories використовується для ведення списку усіх категорій товарів для кращої систематизації списку товарів. Дозволяє додавати та видаляти категорії товарів.

Форма Personel використовується для контролю за штатом працівників. Дозволяє додавати нових працівників з інформацією про їх імена, посади та контактну інформацію. Також форма виконує редагування та видалення вже існуючих у базі даних робітників. Дозволяє виконувати пошук необхідного співробітника за різними критеріями.

Форма OrderingHistory виконує роль архіву продажів за весь час з точною датою, тим самим вона дозволяє полегшити облік при інвентаризації товарів.

Форма Ordering є однією з основних форм програмного застосування, вона виконує функцію оформлення замовлення, відправки завершеного замовлення до архіву продажів і формує товарний чек.

Форми Check та PriceList є додатковими формами для виведення на друк товарного чека та прайс-листа.

Для зв'язку з базою даних на основі ядра Microsoft Jet був використаний провайдер OLE DB.

OLE DB визначає набір COM-інтерфейсів для надання програмам одноманітного доступу до даних, що зберігаються в різних джерелах інформації. Такий підхід дозволяє джерелу даних ділитися своїми даними через інтерфейси, які підтримують обсяг функцій СУБД, які підходять для джерела даних. За своєю розробкою високопродуктивна архітектура OLE DB ґрунтується на використанні гнучкої моделі служб на основі компонентів. Замість того, щоб мати певну кількість проміжних рівнів між програмою та даними, OLE DB вимагає лише стільки компонентів, скільки потрібно для виконання конкретного завдання [10].

Основний функціонал, такі як додавання, редагування, видалення та пошук, реалізований за допомогою SQL запитів до бази даних. Оскільки ми використовуємо технологію ADO.NET, для кожного нового класу ми повинні підключатись до бази даних, адже технологія ADO.NET дозволяє використовувати базу даних лише коли вона потрібна, тим самим ми підвищуємо швидкість програми. Для підключення до бази даних використовуємо конструкцію представлену на рис. 3.2.

```
string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data Source=Data.mdb";  
OleDbConnection connection = new OleDbConnection();  
connection.ConnectionString = ConnectionString;
```

Рисунок 3.2 – Підключення до бази даних

Для виведення даних з бази даних, використовуємо вбудований елемент DataGridView. Виведення даних на DataGridView показано на рис. 3.3.

```
string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data Source=Data.mdb";
OleDbConnection connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;

string sql = "SELECT * FROM Товари";
OleDbCommand myCommand = new OleDbCommand(sql, connection);
connection.Open();

OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 400;
connection.Close();
```

Рисунок 3.3 – Виведення даних з БД на DataGridView

Для реалізації функції виведення на друк чеку та прайс-листа був використаний елемент ReportViewer, передбачений в системі Microsoft Visual Studio. Цей елемент управління здійснює зв'язок додатку (програми) з файлом звіту формату.rdlc. Спочатку потрібно перейти у режим проектування форми. Далі, з панелі інструментів ToolBox у вкладці Reporting потрібно вибрати елемент управління ReportViewer і розмістити його на окремій формі. У результаті система автоматично запропонує вибрати файл звіту у вікні ReportViewer Tasks. Крім того, для елемента управління буде сформовано екземпляр (об'єкт) [16].

Після створення репорту, використовуючи вбудовані елементи будуюмо звіт. Вигляд звіту для чека і прайс-листа зображені на рис 3.4 – 3.5.

Прайс - Лист						[&ExecutionTime]	
Код Товару	Назва товару			Код Категорії	Ціна товару	Виробник	На складі
[КодТовару]	[Назва]			[КодКатегорії]	[ЦінаТовару]	[Виробник]	[Sum(Залишок

Рисунок 3.4 – Вигляд прайс-листа

[&ExecutionTime]		Товарний чек		
Назва	Кількість	Ціна Товара	Продавець	Сума по товару
[Назва]	[Кількість]	[ЦінаТовару]	[КодПерсоналу]	[Sum(ЦінаПродажу)]
				Сума «Ехрг»

Рисунок 3.5 – Вигляд товарного чека

Приклади розроблених запитів для керування базою даних наведені у табл. 3.1. Запити, які виконують однакові функції дублюватись до таблиці не будуть.

Таблиця 3.1. Приклади запитів керування базою даних.

№	SQL запит	Функція
1	INSERT INTO Товари (Назва, КодКатегорії, ЦінаТовару, Виробник, ЗалишокНаСкладі)" + "VALUES (" + Name + ", " + id_category + ", " + price + ", " + manufacturer + ", " + count + "")"	Додавання товару до бази даних
2	"Update Товари SET " + " Назва = " + Name + ", " + "КодКатегорії = " + id_category + ", " + "ЦінаТовару = " + price + ", " + "Виробник = " + manufacturer + ", " + "ЗалишокНаСкладі = " + count + " " Where КодТовару = " + id_tovar;	Редагування існуючого товару у базі
3	"DELETE * FROM Товари WHERE КодТовару = " + id_Tovar	Видалення товару
4	"SELECT * From Товари WHERE " + kriteriy + " LIKE" + " '%" + znachennya + "%"	Пошук товару за обраним критерієм

Продовження таблиці 3.1.

№	SQL запит	Функція
5	"INSERT INTO Продажа (НомерЧека, Дата, КодТовару, Назва, ЦінаТовару, Кількість, КодПерсоналу, ЦінаПродажу)" + "VALUES (" + nomer + ", " + date1 + ", " + kodTov1 + ", " + name1 + ", " + price1 + ", " + count1 + ", " + worker1 + ", " + sum1 + ")"	Перенесення проданого товару до архіву продаж.

3.2 Створення користувацького інтерфейсу

Інтерфейс користувача — засіб зручної взаємодії користувача з інформаційною системою. Сукупність засобів для обробки та відбиття інформації, якнайбільше пристосованих для зручності користувача; у графічних системах інтерфейс користувача, втілюється багатовіконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їх розташуванням, сортуванням елементів вікон, гнучкими налагодженнями як самих вікон, так і окремих їх елементів (файли, теки, ярлики, шрифти тощо), доступністю багатокористувацьких налаштувань [15].

Однією з поставлених задач є створення дружнього до користувача інтефейсу з простим та ергономічним дизайном. Інтерфейс не повинен бути надто перевантаженим. Користувач повинен бачити ту частину інтерфейсу яка потрібна йому на даний момент. Не слід використовувати непок'єднувані або малопок'єднувані кольори при розробці дизайну, адже це пок'єршує концентрацію користувача і може пок'єгано впливати на його зір. Тому для програмного застосунку були використані сірі кольори, які не відволікають від роботи і покращують процес читання під час довгої роботи.

Елементи кожного вікна програми були розбиті на окремі групи в залежності від обраної дії, це зроблено для того, щоб користувач не губився у

інтерфейсі, коли обрана конкретна дія, така як додавання товару, інші елементи будуть приховані від користувача. Вікна товарів, категорій та персоналу при відкритті відображають лише списки, наступні дії обираються за допомогою верхнього меню як це представлено на рис. 3.6.

Додати Редагувати Видалити Пошук				
	КодПерсоналу	ПІБ	Посада	Телефон
►	1	Гулак Владислав Максимович	Директор	380635733758
	2	Іванов Іван Іванович	Менеджер	380501648532
	3	Дія Надія Володимирівна	Старший продавець	380508762532
	4	Сергієнко Володимир Віталійович	Продавець	380631687529
*				

Рисунок 3.6 – Вигляд вікна персоналу при відкритті

3.3 Висновки за розділом 3

У розділі 3 була сформована структура функціоналу програмного продукту, на основі яких були розглянуті рішення щодо реалізації основного функціоналу програми. Було показано як працювати з базою даних за допомогою технології ADO.NET, використовувати інструментарій для створення звітів.

Розглянуто питання щодо реалізації інтерфейсу. На основі загальних потреб користувача до користувацького інтерфейсу був створений інтуїтивно зрозумілий та простий інтерфейс, з мінімальним навантаженням на екрані, задля підвищення комфорту та продуктивності праці.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Умови застосування та призначення програмного продукту

Програму створено для управління магазином і обліку товару на персональному комп'ютері. Оформлення продажів та друку чеків. Основною перевагою є використання ПК замість дорогих POS – систем.

Програма виконує наступні функції:

- відображення всіх наявних товарів у магазини;
- додавання нових товарів до бази даних;
- редагування вже існуючих у базі товарів
- видалення товарів з бази даних;
- пошук необхідного товару за різними критеріями;
- ведення залишку товару на складі;
- додавання, редагування, видалення працівників з бази;
- пошук необхідного працівника у базі;
- перегляд інформації по вже обробленим замовленням з указанням дати та суми замовлення;
- оформлення замовлення покупцю;
- друк товарного чека.

Для користування програмним продуктом користувач повинен мати:

- ПК на базі двоядерного процесора AMD або Intel з тактовою частотою 2.0 GHz та вище;
- 2 ГБ оперативної пам'яті або вище;
- інтегрована або дискретка система відеовиводу;
- жорсткий диск або твердотілий накопичувач об'ємом від 256 ГБ;
- SVGA – монітор;
- клавіатура та маніпулятор типу «миша»;
- для друку звітів необхідний принтер.

Умови до програмної частини:

- на ПК встановлена операційна система MS Windows 7 або вище;
- на ПК встановлено розповсюджуваний пакет Microsoft Report Viewer 2015 Runtime;
- встановлений пакет MS .NET Framework.

Функціональні характеристики повністю задовольняють вимогам описаним у додатку А, текст програми наведено у додатку Б.

4.2. Комплект постачання програмного продукту

Програма розповсюджується за допомогою хмарного сховища або зовнішніх носіїв, в залежності від технічних можливостей користувача.

До комплекту поставки входить саморозпаковуючий архів до якого входить .exe файл програми, файл бази даних у форматі .mdb і повна технічна документація. При оновлені програмного застосунку клієнта буде повідомлено, і вислано нову версію програми.

4.3 Характеристика програми

Програмний застосунок виконаний мовою C#, з використанням .NET Framework у середовищі розробки Microsoft VisualStudio 2022 Community.

База даних керується системою управління базами даних Microsoft Access. Програма автоматично, без участі користувача зберігає всі його дії.

4.4 Інструкція з експлуатації програми

4.4.1 Інсталяція програмного застосунку

Програмний продукт поставляється у вигляді саморозпаковуючого архіву. Для встановлення необхідно зробити подвійний клік по іконці архіву, обрати місце розпаковки, і натиснути «Ок». Після завершення розпаковки ви побачите структуру файлів зображену на рис. 4.1.

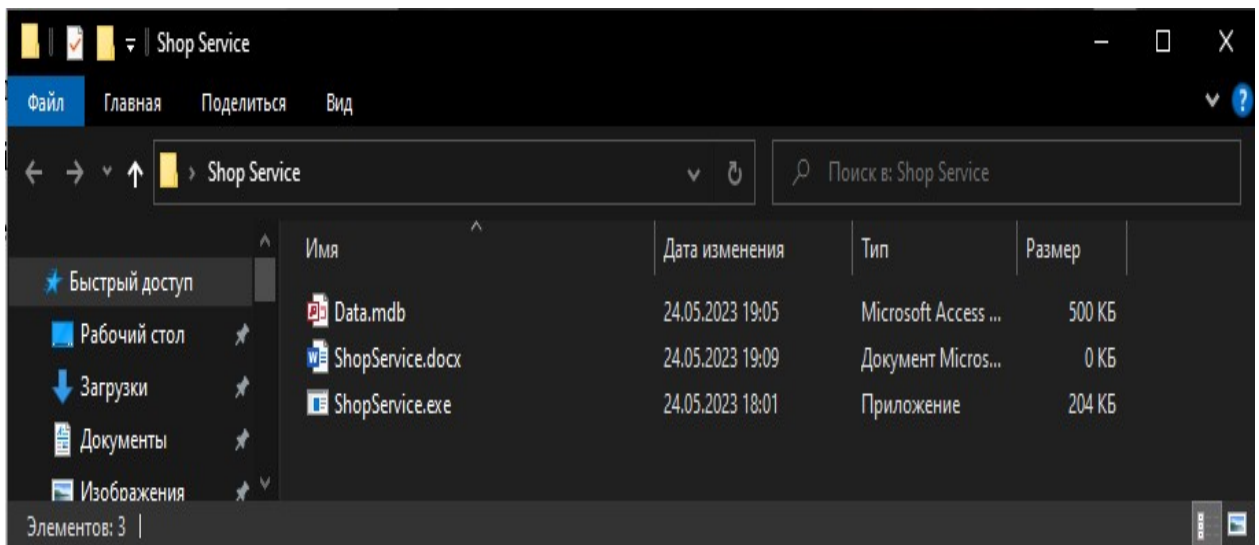


Рисунок 4.1 – Структура файлів інсталюваної програми

Після розпаковки програма повністю готова до роботи.

4.4.2 Робота з програмою

Для початку роботи з програмою зробіть подвійний клік по ярлику ShopService. Після цього ви потрапите на головне вікно програми.

Вигляд головного вікна представлений на рис. 4.2.

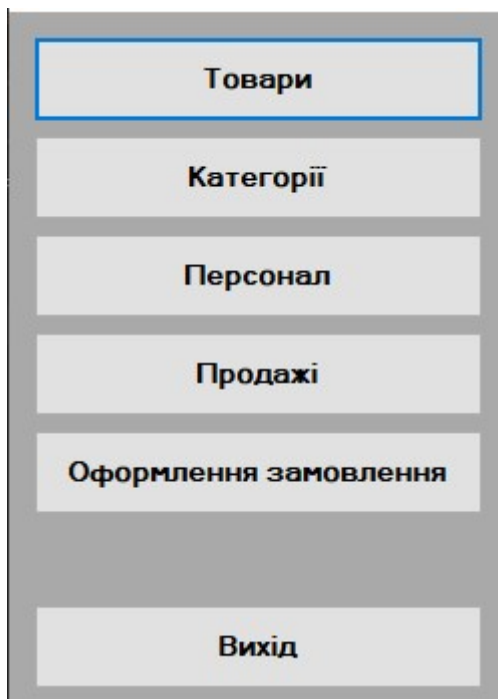


Рисунок 4.2 – Головне вікно програми

4.4.2.1 Робота з товарами

Натиснувши кнопку «Товари» ви перейдете до вікна обліку товарів. Вигляд вікна представлений на рис. 4.3.

Дії Пошук						
	КодТовару	Назва	КодКатегорії	ЦінаТовару	Виробник	ЗалишокНаСкладі
►	11	Спінінг Feima Mirah 2.1 м (5-20r)	2	387	Mirah	1
	12	Спінінг Телескопічний Feima Alivio Slim TE GT 2.4м (15-40r)	2	664	Feima	3
	13	Лісочка Carp Fluoro Mono Orange 300m 0.25mm - 9.3кг	2	199	CarpFluo	10
	14	Волосін Mifine Male 3D Line 600 м Ø0.25 мм	2	376	Mifine	2
	15	Волосін Jaxon Uran Power 0.25 mm 125 m	2	80	Jaxon	100
	16	Шнур плетений Intech First Braid X4 Orange 100m 0.09 мм (6lb/2.72kg)	3	155	Intech	10
	17	Шнур Select Basic PE 150m 0.06mm 6lb/3kg (темно-зелений)	3	255	Select	15
	1	Махове вудилище Brain Phantom Polo 5m, 15-40g	1	1200	Brain	10
	2	Спінінгове вудилище Calipso Blaster Black 3.5m 5-10g	2	600	Calipso	3
	7	Вудилище Jaxon Float Academy Bolo Tele Bream 5-35g	1	1850	Jaxon	9
	8	Болонське вудилище Brain Phantom Bolo 5m 15-40g	1	2780	Brain	3
	9	Болонське вудилище Brain Classic Bolo 5m 10-30g	1	834	Brain	5
	10	Махове вудилище Weida Titan Lake Pro 5 м (до 45 r)	1	758	Weida	1
*						

Рисунок 4.3. – Вікно Товари

Це вікно призначене для обліку наявних товарів на складі. В якості тестової бази даних використовується база даних для риболовного магазину. У даному вікні ви можете виконувати приход товару (додавання), списання (видалення), редагування вже існуючих товарів, та виконувати пошук необхідного товару за трьома критеріями:

- код товару;
- виробник;
- категорія.

Щоб відкрити панель додавання товару натисніть на пункт меню «Дії» та оберіть підпункт «Додати». Заповніть всі пусті поля та натисніть кнопку «Додати товар», у разі виникнення потреби сказувати додавання товару передбачено кнопку «Скасувати». Додавання нового товару зображено на рис. 4.4.

Дії
Пошук

	КодТовару	Назва	КодКатегорії	ЦінаТовару	Виробник	ЗалишокНаСкладі
	11	Спінінг Feima Mirah 2.1 м (5-20г)	2	387	Mirah	1
	12	Спінінг Телескопічний Feima Alivio Slim TE GT 2.4м (15-40г)	2	664	Feima	3
	13	Лісочка Carp Fluoro Mono Orange 300m 0.25мм - 9.3kg	2	199	CarpFluo	10
	14	Волосінь Mifine Male 3D Line 600 м Ø0.25 мм	2	376	Mifine	2
	15	Волосінь Jaxon Uran Power 0.25 mm 125 m	2	80	Jaxon	100
	16	Шнур плетений Intech First Braid X4 Orange 100m 0.09 мм (6lb/2.72kg)	3	155	Intech	10
	17	Шнур Select Basic PE 150m 0.06mm 6lb/3kg (темно-зелений)	3	255	Select	15
	1	Махове вудилище Brain Phantom Polo 5m, 15-40g	1	1200	Brain	10
	2	Спінінгове вудилище Calipso Blaster Black 3.5m 5-10g	2	600	Calipso	3
	7	Вудилище Jaxon Float Academy Bolo Tele Bream 5-35g	1	1850	Jaxon	9
	8	Болонське вудилище Brain Phantom Bolo 5m 15-40g	1	2780	Brain	3
	9	Болонське вудилище Brain Classic Bolo 5m 10-30g	1	834	Brain	5
▶	10	Махове вудилище Weida Titan Lake Pro 5 м (до 45 г)	1	758	Weida	1
*						

Назва
Котушка Weida CHALLENGER XA 10000 (7+1 BB)
Код категорії
5
Ціна товару
1680
Виробник
Weida
На складі
2

Додати товар
Скасувати

Рисунок 4.4 – Додавання товару до бази

Для видалення товару зі списку натисніть на пункт меню «Дії» та оберіть підпункт «Видалити». Після цього ви можете обрати товар на видалення або ввівши код товару на панель видалення, або натиснувши на необхідний товар у таблиці.

Для полегшення пошуку необхідного товару на видалення ви можете комбінувати видалення товару з пошуком, для цього оберіть пункт меню «Пошук».

Видалення товару та пошук зображено на рис. 4.5.

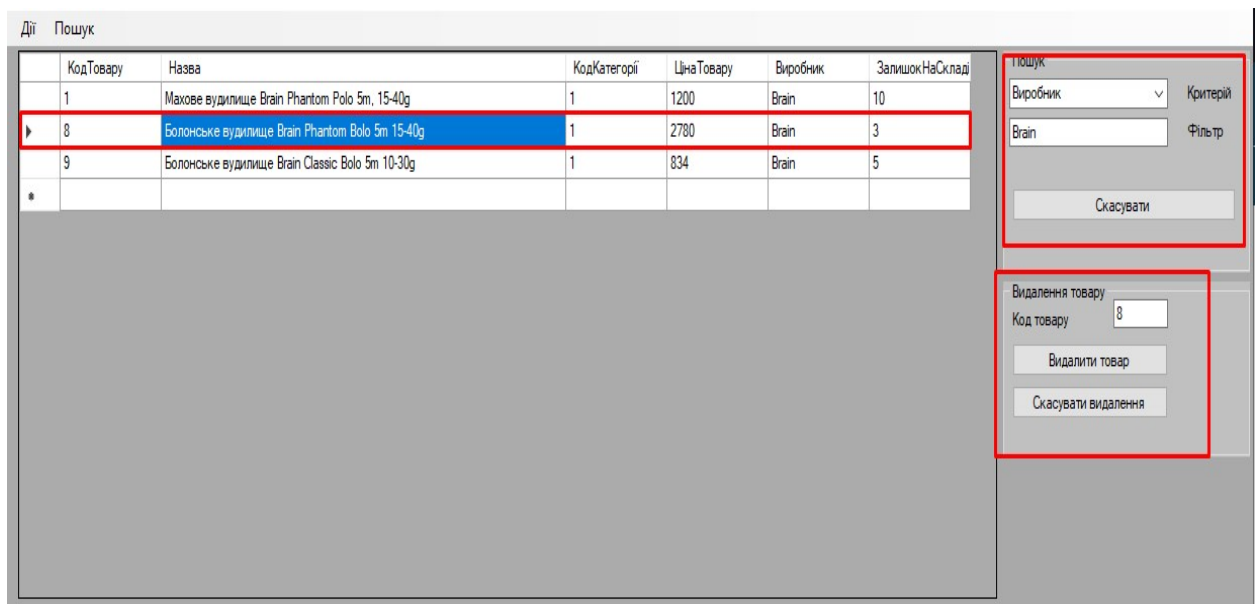


Рисунок 4.5 – Видалення та пошук товару

Для редагування товару натисніть на пункт меню «Дії» та оберіть підпункт «Редагувати». Далі оберіть товар у таблиці (ви так само можете комбінувати з пошуком), Відредагуйте необхідні поля і натисніть «Редагувати». Редагування представлено рис. 4.6.

Дії Пошук

	КодТовару	Назва	КодКатегорії	ЦінаТовару	Виробник	ЗалишокНаСкладі
	11	Спінінг Feima Mirah 2.1 м (5-20r)	2	387	Mirah	1
	12	Спінінг Телескопічний Feima Alivio Slim TE GT 2.4м (15-40r)	2	664	Feima	3
	13	Лісочка Carp Fluoro Mono Orange 300m 0.25mm - 9.3kg	2	199	CarpFluo	10
	14	Волосінь Mifine Male 3D Line 600 м Ø0.25 мм	2	376	Mifine	2
	15	Волосінь Jaxon Uran Power 0.25 mm 125 m	2	80	Jaxon	100
	16	Шнур плетений Intech First Braid X4 Orange 100m 0.09 мм (6lb/2.72kg)	3	155	Intech	10
	17	Шнур Select Basic PE 150m 0.06mm 6lb/3kg (темно-зелений)	3	255	Select	15
	18	Котушка Weida CHALLENGER XA 10000 (7+1 BB)	5	1680	Weida	2
	1	Махове вудилище Brain Phantom Polo 5m, 15-40g	1	1200	Brain	10
	2	Спінінгове вудилище Calipso Blaster Black 3.5m 5-10g	2	600	Calipso	3
	7	Вудилище Jaxon Float Academy Bolo Tele Bream 5-35g	1	1850	Jaxon	9
	8	Болонське вудилище Brain Phantom Bolo 5m 15-40g	1	2780	Brain	3
►	9	Болонське вудилище Brain Classic Bolo 5m 10-30g	1	834	Brain	5
	10	Махове вудилище Weida Titan Lake Pro 5 м (до 45 r)	1	758	Weida	1
*						

Назва: Болонське вудилище Brain Classic Bolo 5m 10-30g
Код категорії: 1
Ціна товару: 834
Виробник: Brain
На складі: 10

Редагувати
Скасувати

Рисунок 4.6 – Редагування товару

Пошук був описаний у видаленні та редагуванні. Під час пошуку товару вам не треба вводити повністю те що ви хочете знайти, програма почне пошук в автоматичному режимі, як тільки ви введете першу букву у строку пошуку. Робота пошуку представлена на рис. 4.5.

4.4.2.2 Робота з категоріями товарів

Натиснувши на кнопку «Категорії» на головному вікні, ви перейдете до вікна роботи з категоріями товарів. В ній зберігається інформація про всі наявні у магазині категорії товарів. Вигляд вікна категорій представлений на рис. 4.7.

Додати Видалити	
КодКатегорії ▲	НазваКатегорії
1	Поплавочні вудилища
2	Спінінгові вудилища
3	Волосінь
4	Плетений шнур
5	Котушки
6	Силіконові приманки
7	Металічні приманки
8	Грузила
9	Повідці
10	Карпові вудилища
11	Род-поди
12	Сигналізатори
13	Прикормки
14	Живі насадки
15	Штучні насадки
16	Поплавці

Рисунок 4.7. – Вигляд вікна категорій

В цьому вікні передбачено дві функції:

- додавання нової категорії;
- видалення вже існуючої категорії.

Щоб додати категорію натисніть на пункт меню «Додати» у поле, що з'явилося введіть назву нової категорії, та натисніть кнопку «Додати».

Додавання нової категорії представлено на рис. 4.8.

КодКатегорії	НазваКатегорії
1	Поплавочні вудилища
2	Спінінгові вудилища
3	Волосінь
4	Плетений шнур
5	Котушки
6	Силіконові приманки
7	Металічні приманки
8	Грузила
9	Повідці
10	Карпові вудилища
11	Род-поди
12	Сигналізатори
13	Прикормки
14	Живі насадки
15	Штучні насадки
16	Поплавці

Назва категорії:

Рисунок 4.8 – Додавання категорії

Видалення категорії аналогічне видаленню товару і не потребує повторного згадування.

4.4.2.3 Робота з персоналом

Для роботи з персоналом на головному вікні натисніть кнопку «Персонал».

Вікно персоналу призначене для контролю за працівниками, відображення їх даних.

Зовнішній вигляд представлений на рис. 4.9.

Додати Редагувати Видалити Пошук				
	КодПерсоналу	ПІБ	Посада	Телефон
▶	1	Гулак Владислав Максимович	Директор	380635733758
	2	Іванов Іван Іванович	Старший продавець	380502643851
	3	Дія Надія Володимирівна	Продавець	380634851672
*				

Рисунок 4.9 – Зовнішній вигляд вікна персоналу

Вікно персонал має аналогічний функціонал: додавання, редагування, видалення та пошук. Модулі взаємодії розроблені аналогічними за інтерфейсом для пришвидшення звикання до програмного продукту. Функціонал вікна персоналу зображений на рис. 4.10 – 4.12.

Додати Редагувати Видалити Пошук				
	КодПерсоналу	ПІБ	Посада	Телефон
	1	Гулак Владислав Максимович	Директор	380635733758
	2	Іванов Іван Іванович	Старший продавець	380502643851
▶	3	Дія Надія Володимирівна	Продавець	380634851672
*				

Додавання робітника					
П.І.Б.	Синенко Владислав Петрович			Додати	
Посада	Продавець				
Контактная інформація	380634851672				
				Скасувати	

Рисунок 4.10 – Додавання нового співробітника

Додати Редагувати Видалити Пошук

	КодПерсоналу	ПІБ	Посада	Телефон
	1	Гулак Владислав Максимович	Директор	380635733758
	2	Іванов Іван Іванович	Старший продавець	380502643851
▶	3	Дія Надія Володимирівна	Продавець	380634851672
*				

Редагування списку персонала

П.І.Б.

Посада

Контактна інформація

Код співробітника

Рисунок 4.11 – Редагування запису

Додати Редагувати Видалити Пошук

	КодПерсоналу	ПІБ	Посада	Телефон
	2	Іванов Іван Іванович	Старший продавець	380502643851
▶	3	Дія Надія Володимирівна	Продавець	380634851672
*				

Видалення

Код співробітника

Пошук

Посада Критерій

про Фільтр

Рисунок 4.12 – Видалення та пошук

4.4.2.4 Оформлення замовлення та створення чека

Для оформлення замовлення перейдіть до вікна оформлення замовлень, натиснувши на кнопку «Оформлення замовлення» на головній формі. Вигляд вікна представлено на рис. 4.13.

КодТовару	Назва	КодКатегорії	ЦінаТовару	Виробник	ЗалишокНаСкладі
1	Макове вудлище Brain Phantom Polo 5m, 15-40g	1	1200	Brain	10
2	Слінгове вудлище Calproo Blaster Black 3.5m 5-10g	2	600	Calproo	3
7	Вудлище Jaxon Float Academy Bolo Tele Bream 5-35g	1	1850	Jaxon	9
8	Болонське вудлище Brain Phantom Bolo 5m 15-40g	1	2780	Brain	3
9	Болонське вудлище Brain Classic Bolo 5m 10-30g	1	834	Brain	10
10	Макове вудлище Weida Titan Lake Pro 5 m (до 45 г)	1	758	Weida	1
11	Слінг Feima Mirah 2.1 m (5-20г)	2	387	Mirah	1
12	Слінг Телескопичний Feima Alivio Slim TE GT 2.4m (15-40г)	2	664	Feima	3
13	Пісочка Carp Fluo Mono Orange 300m 0.25mm - 9.3кг	2	199	CarpFluo	10

КодПерсоналу	ПІБ	Посада	Телефон
1	Гулак Владисла...	Директор	380635733758
2	Іванов Іван Іван...	Старший прода...	380502643851
3	Діа Надія Волод...	Продавець	380634851672

ЧЕК

КодЧека	Дата	КодТовару	Назва	ЦінаТовару	Кількість	КодПерсоналу	ЦінаПродажу
4	24.05.2023	1	Макове вудлище Brain Phantom Polo 5m, 15-40g	1200	1	1	1200
5	24.05.2023	2	Слінгове вудлище Calproo Blaster Black 3.5m 5-10g	600	1	1	600
6	24.05.2023	2	Слінгове вудлище Calproo Blaster Black 3.5m 5-10g	600	1	1	600

Відкрити чек Прайс - Лист

Рисунок 4.13 – Вигляд вікна оформлення замовлень

У вікні ви побачите три таблиці:

- таблицю товарів;
- таблицю персоналу;
- таблицю «Чек»

Таблиця «Чек» у цьому вікні зберігає дані про останню покупку до моменту відкриття нового чеку. Після відкриття нового чеку дані автоматично відправляється до архіву продажів, до якого ми повернемося пізніше.

Для створення нового чеку потрібно натиснути на кнопку «Відкрити чек». Після натискання спочатку буде виведена підказка до роботи яка зображена на рис. 4.14.

Виберіть своє ім'я, оберіть потрібний товар. Після цього оберіть кількість товару та натисніть кнопку [Розрахувати]. Після цього натисніть кнопку [Записати у чек]

OK

Рисунок 4.14 – Підказка до роботи

Після цього вам потрібно обрати необхідний товар, ввести кількість та натиснути кнопку «Розрахувати», після цього підіб'ється сума то товару.

Далі обираємо у таблиці запис про продавця і натискаємо «Записати в чек». Готово, ми додали перший запис до чека. Далі продовжуємо заповнювати чек доки не вкажете усі необхідні товари, по завершенню натисніть на кнопку «Друк», і перед вами з'явиться товарний чек, вже готовий до друку, натисніть кнопку «Друк» у чеку для роздрукування.

При необхідності провести операцію продажу за минулим часом (якщо наприклад вимкнули світло), ви можете змінити дату продажу натиснувши на стрілочку у правому боці поля з датою.

У вікні «Чек» можна вивести прайс–лист, який буде готовий до друку, процес роботи аналогічний чеку.

Вікно оформлення замовлення представлено на рис. 4.15.

КодТовару	Назва	КодКатегорії	ЦінаТовару	Виробник	ЗалишокНаСклад
11	Стінні Feima Mirah 2.1 м (5-20r)	2	387	Mirah	1
12	Стінні Телевізійний Feima Alvio Slim TE GT 2.4м (15-40r)	2	664	Feima	3
13	Пісочка Cap Fluo Mono Orange 300m 0.25mm - 9.3kr	2	199	CapFluo	10
14	Волокно Mline Male 3D Line 600 м 00.25 мм	2	376	Mline	2
15	Волокно Jaxon Utran Power 0.25 mm 125 m	2	80	Jaxon	100
16	Шнур плетений Intech First Braid X4 Orange 100m 0.09 мм (8b/2.72kg)	3	155	Intech	10
17	Шнур Select Basic PE 150m 0.09mm 8b/3kg (темно-зелений)	3	255	Select	15
18	Котушка Welda CHALLENGER XA 10000 (7+1 BB)	5	1680	Welda	1
1	Максое видилище Brain Phantom Polo 5m, 15-40g	1	1200	Brain	10

КодПерсоналу	ПІБ	Посада	Телефон
1	Гулак Владисла...	Директор	38063573750
2	Іванюк Іван Іван...	Старший продавець	380502643851
3	Дія Надія Волод...	Продавець	380634851672

Оформлення замовлення

серед , 24 мая 2023 г.

Дата

1

Кількість

Шнур Select Basic PE 150m 0.09mm 8b/3kg

17

Товар

Іванюк Іван Іванович

2

Продавець

255

Ціна товару

255

Розрахувати

Сума

Записати в чек

Друк

ЧЕК

КодЧека	Дата	КодТовару	Назва	ЦінаТовару	Кількість	КодПерсоналу	ЦінаПродажу
7	24.05.2023	8	Болонське видилище Brain Phantom Polo 5m 15-40g	2780	1	2	2780
8	24.05.2023	18	Котушка Welda CHALLENGER XA 10000 (7+1 BB)	1680	1	2	1680

Відкрити чек

Прайс - Лист

Рисунок 4.15 – Вигляд вікна та створення чека

4.4.2.5 Архів замовлень

Для переходу до архіву натисніть на кнопку «Продажи». Він призначений для збереження інформації про всі продані товари, яку можна використати для інвентаризацій, або розрахунків прибутків. Вигляд вікна представлений на рис. 4.16.

	КодПродажу	НомерЧека	Дата	КодТовару	Назва	ЦінаТовару	Кількість	КодПерсоналу	ЦінаПродажу
►	20	22	02.05.2023	12	Спінінг Телеско...	664	1	2	664
	21	22	02.05.2023	17	Шнур Select Basi...	255	2	3	510
	22	22	02.05.2023	10	Махове вудили...	758	1	1	758
	23	22	17.05.2023	7	Вудилище Жакоп...	1850	1	1	1850
	24	22	17.05.2023	13	Лісочка Carp Flu...	199	2	3	398
	25	22	24.05.2023	17	Шнур Select Basi...	255	1	3	255
	26	22	23.05.2023	9	Болонське вуди...	834	2	3	1668
*									

Рисунок 4.16 – Архів продажів

4.5 Тестування програми

При роботі над програмним продуктом було проведене тестування на надійність та стабільність системи. Також було проведено тестування на уникнення вийняткових ситуацій.

Вийняткові ситуації:

- користувач не обрав критерій пошуку і почав заповнювати фільтр;
- користувач не обрав товар, категорію. Або співробітника для видалення.

Якщо при видаленні не вказати товар, категорію чи працівника, програма виведе сповіщення, що зображене на рис. 4.17.

Виберіть строку для видалення



Рисунок 4.17 – Попередження користувача при помилці видалення

В ситуації при якій користувач не обрав критерій пошуку то при введенні необхідного слова програма також виведе попередження, яке зображено на рис. 4.18.

Оберіть критерій пошуку



Рисунок 4.18 – Попередження користувача при помилці пошуку

Якісне тестування програми було проведено за наступними характеристиками:

- функціональність – програмний застосунок здатний вирішувати поставлену задачу, видати потрібний результат;
- надійність – програма підтримує працездатність у відповідних для неї умовах
- практичність – завдяки простому та дружньому інтерфейсу програма не складна у навчанні і зручна для тривалого використання, вона зрозуміла і при роботі з нею не виникають протиріччя;
- продуктивність – при заданих умовах програмний засіб виконує поставлений обсяг задач.

Даний програмний продукт повністю відповідає таким критеріям стабільності:

- взаємодія між внутрішнім модулями;

- стабільна робота бази даних;
- правильне відображення вкладок та їх елементів.

4.6 Висновки за розділом 4

При роботі над розділом 4 були визначені умови роботи створеного програмного засобу, його остаточний функціонал. Розглянули характеристики створеного програмного продукту, також був розглянутий комплект поставки створеної програми. Було розроблено детальне керівництво користувача, з вказанням кожного аспекту роботи з програмою. Проведено тестування системи за різними критеріями.

В результаті було отриманий готовий до використання програмний засіб.

ВИСНОВКИ

У процесі виконання дипломної кваліфікаційної роботи бакалавра було створено програмний застосунок для управління магазином.

Проаналізувавши проблеми та тенденції у сфері торгівлі, провівши аналіз вже існуючих аналогічних програмних рішень були сформовані вимоги до створюваного програмного продукту, його функціонал та направленість на конкретний тип підприємництва такого як мала роздрібна торгівля.

Визначено, що для виконання поставленої задачі необхідно розробити програмний застосунок для управління майже за всіма аспектами роботи у магазині. Користувач отримає програмний продукт який дозволяє вести детальний облік за наявним товаром, працівниками, вести облік проданих товарів і друкувати товарні чеки для покупців, і все це без використання дорогого обладнання.

Для створення програми було використано об'єктно – орієнтовану мову програмування C# з використанням платформи .NET Framework. В якості середовища розробки було використано Microsoft Visual Studio/

Для роботи з даними було використано систему управління базами даних Microsoft Access.

Розроблено структуру функціоналу програмного застосунку на основі якої було виконано всі поставлені задачі. Розроблена програма має простий і зрозумілий користувацький інтерфейс, програма зручна для використання, роботі з нею легко вчитись.

Систему протестовано за різними критеріями, в результаті чого отримано стабільну готову до роботи програму.

Розроблений продукт реалізував всі поставлені задачі дипломної кваліфікаційної роботи бакалавра.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Крамниця [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Крамниця>.
2. Компанія Unipro [Електрон. ресурс]. – Режим доступу: <https://unipro.com.ua/ua/magazyn--market/>.
3. Компанія Zipru Склад [Електрон. ресурс]. – Режим доступу <https://zipru.com.ua/zstore>.
4. C# [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/C_Sharp.
5. C#: що це за мова та де її використовують [Електрон. ресурс]. – Режим доступу: <https://robotdreams.cc/uk/blog/284-s-hto-eto-za-yazyk-i-gde-ego-ispolzuuyut>.
6. .NET Framework [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/.NET_Framework#.NET_Framework_4.0.
7. Microsoft Visual Studio [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio.
8. SQL [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/SQL>.
9. Microsoft Access [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Access.
10. OLE DB providers [Електрон. ресурс]. – Режим доступу: <https://learn.microsoft.com/uk-ua/office/client-developer/access/desktop-database-reference/ole-db-providers>.
11. UML [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Unified_Modeling_Language.
12. Діаграма прецедентів [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Діаграма_прецедентів.

13. Що таке C#? Чи підходить мені ця мова програмування? Чому вона крута? [Електрон. ресурс]. – Режим доступу: <https://beetroot.academy/blog/courses/what-is-C>.
14. Система управління базами даних [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Система_управління_базами_даних.
15. Інтерфейс користувача [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Інтерфейс_користувача.
16. Приклад створення звіту в додатку. Елемент управління Report Viewer. База даних Microsoft Access [Електрон. ресурс]. – Режим доступу: https://www.bestprog.net/uk/2017/12/26/an-example-of-creating-a-report-in-an-application-created-using-the-windows-forms-application-template-the-microsoft-access-database-mdb_ua/#p07_02.

ДОДАТОК А
Технічне завдання

Вступ

Програма, розробка якої ведеться, має назву «система для управління магазином». На сьогоднішній день разом з розвитком технологій, які пришвидшують процеси роботи, почали збільшуватись і обсяги роботи. Сфера торгівлі також не стоїть на місці, розроблено велику кількість різноманітних систем для керування.

Звернувши увагу на дрібних підприємців, ми побачимо, що через фінансові труднощі більшість з них просто не встигають за прогресом, або через небажання навчатись чомусь новому, і на преший погляд дуже складному, або через банальний недостаток коштів. Саме це і стало підставою для вибору цієї теми для її розгляду в ході дипломної роботи бакалавра.

Програмний засіб для управління магазину призначений для покращення якості роботи, пришвидшення процедур обліку і оформлення замовлень.

A.1 Підстави для розробки

Підставою для розробки є виконання завдання дипломної кваліфікаційної роботи бакалавра, за темою «Розробка програмного забезпечення для управління магазином», яку було затверджено наказом № 87 від 04 квітня 2023 року за Національним університетом «Запорізька політехніка»

A.2 Призначення розробки

Створений програмний продукт можна віднести до категорії «Програмне забезпечення». Він створений для покращення якості роботи працівника сфери торгівлі, та пришвидшити облік документів та оформлення замовлень.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціоналу

Програмний продукт повинен виконувати наступні функції:

- відображення всіх наявних товарів у магазини;
- додавання нових товарів до бази даних;
- редагування вже існуючих у базі товарів
- видалення товарів з бази даних;
- пошук необхідного товару за різними критеріями;
- ведення залишку товару на складі;
- додавання, редагування, видалення працівників з бази;
- пошук необхідного працівника у базі;
- перегляд інформації по вже обробленим замовленням з указанням дати та суми замовлення;
- оформлення замовлення покупцю;
- друк товарного чека;

A.3.2 Вимоги до надійності

Система повинна бути стійкою про програмних збоїв, постійно проводити неявне для користувача збереження даних, зберігати інформацію при системних збоях, якщо такі будуть.

А.3.3 Умови експлуатації

Для використання програмного застосунку необхідно мати персональний комп'ютер з підключеним до нього принтером.

А.3.4 Вимоги до складу та параметрів технічних засобів

Для роботи програмного засобу необхідний IBM-сумісний ПЗ на базі процесорів Intel Core 2 Duo або AMD Athlon II x2 з тактовою частотою не менше 2 GHz або вище, об'ємом оперативної пам'яті 2 ГБ та вище, жорсткий диск або твердотільний накопичувач об'ємом від 256 ГБ, SVGA-монітор, маніпулятор типу «миша» та клавіатура. Для виводу на друк потрібен принтер. Операційна система MS Windows 7 та вище зі встановленим розповсюджуваним пакетом Microsoft Report Viewer 2015 Runtime.

А.3.5 Вимоги до маркування та упакування

Програмний засіб може бути записаний на носій або переданий щонад допомогою хмарного сховища. На носії або в підписі хмарного файлу повинно стояти маркування «Shop Service»

А.3.6 Вимоги до зберігання продукту на носіях

Вимоги збереження продукту аналогічні умовам збереження зовнішніх носіїв.

A.4 Етапи розробки

При виконанні роботи заплановано наступні етапи:

- аналіз предметної області;
- аналіз існуючих аналогів;
- розробка вимог до програмного застосунку;
- формування функціоналу системи;
- реалізація програмного застосунку;
- тестування та експлуатація;
- оформлення документації.

A.5 Процес приймання роботи

Виконана згідно за календарним планаом дипломна робота приймається консультантами згідно відповідних розділів. Після чого підписується керівником проєкту та відправляється на перевірку нормоконтролером та системою антиплагіату, після чого вона допускається до захисту завідуючим кафедрою і захищається перед комісією. Роботу необхідно подати рецензенту для отримання рецензії на виконану дипломну роботу.

ДОДАТОК Б
Текст програми

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

```

```

namespace WindowsFormsApplication1

```

```

{
    public partial class Goods : Form
    {
        public int selectrow = -1;
        public Goods()
        {
            InitializeComponent();
        }
    }

```

```

        private void Form2_Load(object sender, EventArgs e)
        {
            this.Height = 450;
            this.Width = 1003;
            string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data
Source=Data.mdb";
            OleDbConnection connection = new OleDbConnection();
            connection.ConnectionString = ConnectionString;

            string sql = "SELECT * FROM Товари";
            OleDbCommand myCommand = new OleDbCommand(sql, connection);

```

```

connection.Open();

OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 400;
connection.Close();
}

private void діїToolStripMenuItem_Click(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string Name = textBox1.Text.ToString();
    string id_category = textBox2.Text.ToString();
    string price = textBox3.Text.ToString();
    string manufacturer = textBox4.Text.ToString();
    string count = textBox5.Text.ToString();

    sql = "INSERT INTO Товари (Назва, КодКатегорії, ЦінаТовару, Виробник,
ЗалишокНаСкладі)" + "VALUES (" + Name + ", " + "\"" + id_category + ", " + "\"" + price + ",
" + "\"" + manufacturer + ", " + "\"" + count + ")";

    myCommand = new OleDbCommand(sql, connection);

```

```

connection.Open();
myCommand.ExecuteNonQuery();
connection.Close();

connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;
sql = "SELECT * From Товари";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
connection.Close();
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
textBox4.Text = "";
textBox5.Text = "";
this.Height = 450;
this.Width = 1003;
}

private void додатиToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Height = 604;
    groupBox1.Visible = true;
    button3.Visible = false;
    button1.Visible = true;
    button2.Visible = true;
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{

```

```

this.Height = 450;
this.Width = 1249;
groupBox2.Visible = true;
groupBox1.Visible = false;
button1.Visible = false;
button2.Visible = false;
button3.Visible = false;

}

private void button4_Click(object sender, EventArgs e)
{
    if (selectrow == -1 || selectrow >= dataGridView1.RowCount - 1)
    {
        MessageBox.Show("Виберіть строку для видалення");
        return;
    }
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string id_Tovar = textBox6.Text.ToString();

    sql = "DELETE * FROM Товари WHERE КодТовару = " + id_Tovar;

    myCommand = new OleDbCommand(sql, connection);
    connection.Open();
    myCommand.ExecuteNonQuery();
    connection.Close();

    connection = new OleDbConnection();

```

```

connection.ConnectionString = ConnectionString;
sql = "SELECT * FROM Товари";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 400;
connection.Close();
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";

    selectrow = dataGridView1.CurrentCell.RowIndex;

    if (selectrow < (dataGridView1.RowCount - 1))
    {
        textBox1.Text = dataGridView1[1, selectrow].Value.ToString();
        textBox2.Text = dataGridView1[2, selectrow].Value.ToString();
        textBox3.Text = dataGridView1[3, selectrow].Value.ToString();
        textBox4.Text = dataGridView1[4, selectrow].Value.ToString();
        textBox5.Text = dataGridView1[5, selectrow].Value.ToString();
        textBox6.Text = dataGridView1[0, selectrow].Value.ToString();
    }
}

```

```

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs
e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";

    selectrow = dataGridView1.CurrentCell.RowIndex;

    if (selectrow < (dataGridView1.RowCount - 1))
    {
        textBox1.Text = dataGridView1[1, selectrow].Value.ToString();
        textBox2.Text = dataGridView1[2, selectrow].Value.ToString();
        textBox3.Text = dataGridView1[3, selectrow].Value.ToString();
        textBox4.Text = dataGridView1[4, selectrow].Value.ToString();
        textBox5.Text = dataGridView1[5, selectrow].Value.ToString();
        textBox6.Text = dataGridView1[0, selectrow].Value.ToString();
    }
}

private void button5_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";
    this.Height = 450;
    this.Width = 1003;
}

```

```

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Height = 604;
    groupBox1.Visible = true;
    button1.Visible = false;
    button2.Visible = true;
    button3.Visible = true;
}

private void button3_Click(object sender, EventArgs e)
{
    if (selectrow == -1 || selectrow >= dataGridView1.RowCount - 1)
    {
        MessageBox.Show("Виділіть строку для редагування");
        return;
    }

    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";

    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string Name = textBox1.Text.ToString();
    string id_category = textBox2.Text.ToString();
    string price = textBox3.Text.ToString();
    string manufacturer = textBox4.Text.ToString();
    string count = textBox5.Text.ToString();
    string id_tovar = textBox6.Text.ToString();

    sql = "Update Товари SET " +
        " Назва = '" + Name + "', " +
        "КодКатегорії = '" + id_category + "', " +

```

```

"ЦінаТовару = "" + price + "," +
"Виробник = "" + manufacturer + "," +
"ЗалишокНаСкладі = "" + count + "" +
" Where КодТовару = " + id_tovar;

```

```

myCommand = new OleDbCommand(sql, connection);
connection.Open();
myCommand.ExecuteNonQuery();
connection.Close();

```

```

connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;
sql = "SELECT * FROM Товари";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 400;
connection.Close();
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    groupBox1.Visible = false;
    this.Height = 450;
    this.Width = 1003;
}

```

```

private void button6_Click(object sender, EventArgs e)
{

}

```

```

private void button7_Click(object sender, EventArgs e)
{
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";

    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;

    sql = "SELECT * From Товари";

    myCommand = new OleDbCommand(sql, connection);
    connection.Open();
    OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
    DataSet ds = new DataSet();
    da.Fill(ds, "Result");
    dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
    connection.Close();
    this.Height = 450;
    this.Width = 1003;
    groupBox3.Visible = false;
}

private void поискToolStripMenuItem_Click(object sender, EventArgs e)
{
    groupBox1.Visible = true;
    groupBox3.Visible = true;
    this.Height = 450;
    this.Width = 1249;
    button1.Visible = false;
    button2.Visible = false;
    button3.Visible = false;
}

```

```

private void textBox7_TextChanged(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex < 0)
    {
        MessageBox.Show("Оберіть критерій пошуку");
        return;
    }
    string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string kriteriy = comboBox1.Text.ToString();
    string znachennya = textBox7.Text.ToString();

    sql = "SELECT * From Товари WHERE " + kriteriy + " LIKE" + " '%" + znachennya +
    "%'";

    myCommand = new OleDbCommand(sql, connection);
    connection.Open();
    OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
    DataSet ds = new DataSet();
    da.Fill(ds, "Result");
    dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
    connection.Close();
}

}

namespace WindowsFormsApplication1
{

```

```

public partial class Categories : Form
{
    public int selectrow = -1;
    public Categories()
    {
        InitializeComponent();
    }

    private void Form6_Load(object sender, EventArgs e)
    {
        string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";

        OleDbConnection connection = new OleDbConnection();
        connection.ConnectionString = ConnectionString;

        string sql = "SELECT * FROM Kateropiï";
        OleDbCommand myCommand = new OleDbCommand(sql, connection);
        connection.Open();

        OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
        DataSet ds = new DataSet();
        da.Fill(ds, "Result");
        dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
        dataGridView1.Columns[1].Width = 400;
        connection.Close();
        this.Width = 385;
        this.Height = 461;
    }

    private void ADDToolStripMenuItem_Click(object sender, EventArgs e)
    {
        panel1.Visible = true;
        groupBox2.Visible = false;
    }

```

```

this.Width = 385;
this.Height = 571;
textBox1.Text = "";
}

```

```

private void DELETEToolStripMenuItem_Click(object sender, EventArgs e)
{
    groupBox2.Visible = true;
    panel1.Visible = false;
    this.Width = 385;
    this.Height = 571;
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    groupBox2.Visible = false;
    this.Width = 385;
    this.Height = 461;
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    panel1.Visible = false;
    this.Width = 385;
    this.Height = 461;
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;

```

```

string sql;
string Name = textBox1.Text.ToString();

sql = "INSERT INTO Kateropii (НазваKateropii)" + "VALUES ('" + Name + "')";

myCommand = new OleDbCommand(sql, connection);
connection.Open();
myCommand.ExecuteNonQuery();
connection.Close();

connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;
sql = "SELECT * From Kateropii";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
connection.Close();
this.Width = 385;
this.Height = 461;
}

private void button4_Click(object sender, EventArgs e)
{
    if (selectrow == -1 || selectrow >= dataGridView1.RowCount - 1)
    {
        MessageBox.Show("Выделите в сетке строку для удаления");
        return;
    }
    string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();

```

```

connection.ConnectionString = ConnectionString;
OleDbCommand myCommand;
string sql;
string id_Tovar = textBox6.Text.ToString();

sql = "DELETE * FROM Kateropii WHERE КодКатегории = " + id_Tovar;

myCommand = new OleDbCommand(sql, connection);
connection.Open();
myCommand.ExecuteNonQuery();
connection.Close();

connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;
sql = "SELECT * FROM Kateropii";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
connection.Close();
this.Width = 385;
this.Height = 461;
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{

    textBox6.Text = "";

    selectrow = dataGridView1.CurrentCell.RowIndex;

```

```

        if (selectrow < (dataGridView1.RowCount - 1))
        {
            textBox6.Text = dataGridView1[0, selectrow].Value.ToString();
        }
    }
}
}
namespace WindowsFormsApplication1
{
    public partial class MainPage : Form
    {
        public MainPage()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Goods f2 = new Goods();
            f2.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Personnel f3 = new Personnel();
            f3.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            OrderingHistory f4 = new OrderingHistory();
            f4.Show();
        }
    }
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    Ordering f5 = new Ordering();
    f5.Show();
}

private void button5_Click(object sender, EventArgs e)
{
    Categories f6 = new Categories();
    f6.Show();
}

private void button6_Click(object sender, EventArgs e)
{
    MessageBoxButtons buttons = MessageBoxButtons.YesNo;
    System.Windows.Forms.DialogResult result;
    result = MessageBox.Show("Ви дійсно бажаєте вийти?", "Вихід", buttons);
    if (result == System.Windows.Forms.DialogResult.Yes)
        Close();
}
}
}

public partial class OrderingHistory : Form
{
    public OrderingHistory()
    {
        InitializeComponent();
    }

    private void Form4_Load(object sender, EventArgs e)
    {
        string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";
        OleDbConnection connection = new OleDbConnection();

```

```

connection.ConnectionString = ConnectionString;

string sql = "SELECT * From Продажа";
OleDbCommand myCommand = new OleDbCommand(sql, connection);
connection.Open();

OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
connection.Close();
}
}
}
public partial class Personnel : Form
{
    public int selectrow = -1;
    public Personnel()
    {
        InitializeComponent();
    }

    private void Form3_Load(object sender, EventArgs e)
    {
        this.Height = 255;
        this.Width = 885;
        string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data
Source=Data.mdb";
        OleDbConnection connection = new OleDbConnection();
        connection.ConnectionString = ConnectionString;
        string sql = "SELECT * From Персонал";
        OleDbCommand myCommand = new OleDbCommand(sql, connection);
        connection.Open();

```

```

OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 300;
dataGridView1.Columns[2].Width = 300;
connection.Close();
}

```

```

private void ADDToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Width = 888;
    this.Height = 421;
    groupBox1.Visible = true;
    groupBox1.Text = "Додавання робітника";
    textBox4.Visible = false;
    button1.Visible = true;
    button3.Visible = false;
    label4.Visible = false;
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string FIO = textBox1.Text.ToString();
    string Position = textBox2.Text.ToString();
    string PhoneNumber = textBox3.Text.ToString();

```

```
sql = "INSERT INTO Персонал (ПИБ, Посада, Телефон)" + "VALUES ('" + FIO + "', "
+ "'" + Position + "', '" + PhoneNumber + "')";
```

```
myCommand = new OleDbCommand(sql, connection);
connection.Open();
myCommand.ExecuteNonQuery();
connection.Close();
```

```
connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;
sql = "SELECT * FROM Персонал";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 300;
dataGridView1.Columns[2].Width = 300;
connection.Close();
this.Height = 255;
this.Width = 885;
```

```
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    groupBox1.Visible = false;
    this.Height = 255;
    this.Width = 885;
```

```
}
```

```
private void EDITToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
```

```
    groupBox1.Visible = true;
```

```
    groupBox1.Text = "Редагування списку персонала";
```

```
    button3.Visible = true;
```

```
    label4.Visible = true;
```

```
    textBox4.Visible = true;
```

```
    textBox4.Enabled = false;
```

```
    button1.Visible = false;
```

```
    this.Width = 888;
```

```
    this.Height = 421;
```

```
}
```

```
private void button3_Click(object sender, EventArgs e)
```

```
{
```

```
    if (selectrow == -1 || selectrow >= dataGridView1.RowCount - 1)
```

```
    {
```

```
        MessageBox.Show("Виділіть спрок для редагування");
```

```
        return;
```

```
    }
```

```
    string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data  
Source=Data.mdb";
```

```
    OleDbConnection connection = new OleDbConnection();
```

```
    connection.ConnectionString = ConnectionString;
```

```
    OleDbCommand myCommand;
```

```
    string sql;
```

```
    string FIO = textBox1.Text.ToString();
```

```
    string Position = textBox2.Text.ToString();
```

```
    string PhoneNumber = textBox3.Text.ToString();
```

```
    string id_Personal = textBox4.Text.ToString();
```

```

sql = "Update Персонал SET " +
      " ПИБ = '" + FIO + "', " +
      "Посада = '" + Position + "', " +
      "Телефон = '" + PhoneNumber + "'" +
      " Where КодПерсоналу = " + id_Personal;

myCommand = new OleDbCommand(sql, connection);
connection.Open();
myCommand.ExecuteNonQuery();
connection.Close();

connection = new OleDbConnection();
connection.ConnectionString = ConnectionString;
sql = "SELECT * FROM Персонал";
myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
dataGridView1.Columns[1].Width = 300;
dataGridView1.Columns[2].Width = 300;
connection.Close();
this.Height = 255;
this.Width = 885;

}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
}

```

```
textBox5.Text = "";
```

```
selectrow = dataGridView1.CurrentCell.RowIndex;
```

```
if(selectrow < (dataGridView1.RowCount - 1))
```

```
{
```

```
    textBox4.Text = dataGridView1[0, selectrow].Value.ToString();
```

```
    textBox1.Text = dataGridView1[1, selectrow].Value.ToString();
```

```
    textBox2.Text = dataGridView1[2, selectrow].Value.ToString();
```

```
    textBox3.Text = dataGridView1[3, selectrow].Value.ToString();
```

```
    textBox5.Text = dataGridView1[0, selectrow].Value.ToString();
```

```
}
```

```
}
```

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs
```

e)

```
{
```

```
    textBox1.Text = "";
```

```
    textBox2.Text = "";
```

```
    textBox3.Text = "";
```

```
    textBox4.Text = "";
```

```
    textBox5.Text = "";
```

```
selectrow = dataGridView1.CurrentCell.RowIndex;
```

```
if (selectrow < (dataGridView1.RowCount - 1))
```

```
{
```

```
    textBox4.Text = dataGridView1[0, selectrow].Value.ToString();
```

```
    textBox1.Text = dataGridView1[1, selectrow].Value.ToString();
```

```
    textBox2.Text = dataGridView1[2, selectrow].Value.ToString();
```

```
    textBox3.Text = dataGridView1[3, selectrow].Value.ToString();
```

```
    textBox5.Text = dataGridView1[0, selectrow].Value.ToString();
```

```
}
```

```

    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (selectrow == -1 || selectrow >= dataGridView1.RowCount - 1)
        {
            MessageBox.Show("Виділіть строку для видалення");
            return;
        }

        string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data
Source=Data.mdb";

        OleDbConnection connection = new OleDbConnection();
        connection.ConnectionString = ConnectionString;
        OleDbCommand myCommand;
        string sql;
        string id_Personal = textBox5.Text.ToString();

        sql = "DELETE * FROM Персонал WHERE КодПерсоналу = " + id_Personal;

        myCommand = new OleDbCommand(sql, connection);
        connection.Open();
        myCommand.ExecuteNonQuery();
        connection.Close();

        connection = new OleDbConnection();
        connection.ConnectionString = ConnectionString;
        sql = "SELECT * FROM Персонал";
        myCommand = new OleDbCommand(sql, connection);
        connection.Open();
        OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
        DataSet ds = new DataSet();
        da.Fill(ds, "Result");
        dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
    }

```

```
dataGridView1.Columns[1].Width = 300;
dataGridView1.Columns[2].Width = 300;
connection.Close();
this.Height = 255;
this.Width = 885;
}

private void DELETEToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Width = 1196;
    this.Height = 255;
    groupBox2.Visible = true;
    groupBox1.Text = "Видалення співробітника";
}

private void button5_Click(object sender, EventArgs e)
{
    textBox5.Text = "";
    groupBox2.Visible = false;
    this.Height = 255;
    this.Width = 885;
}

private void SEARCHToolStripMenuItem_Click(object sender, EventArgs e)
{
    groupBox3.Visible = true;
    this.Width = 1196;
    this.Height = 255;
}

private void button6_Click(object sender, EventArgs e)
```

```

{
    if (comboBox1.SelectedIndex < 0)
    {
        MessageBox.Show("Виберіть критерій пошуку");
        return;
    }
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string kriteriy = comboBox1.Text.ToString();
    string znachennya = textBox6.Text.ToString();

    sql = "SELECT * From Персонал WHERE "
+ kriteriy + " LIKE" + " '" + znachennya + "'";

    myCommand = new OleDbCommand(sql, connection);
    connection.Open();
    OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
    DataSet ds = new DataSet();
    da.Fill(ds, "Result");
    dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
    connection.Close();
}

private void button7_Click(object sender, EventArgs e)
{
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";
    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;

```

```

OleDbCommand myCommand;
string sql;

sql = "SELECT * From Персонал";

myCommand = new OleDbCommand(sql, connection);
connection.Open();
OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
DataSet ds = new DataSet();
da.Fill(ds, "Result");
dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
connection.Close();
}

private void textBox6_TextChanged(object sender, EventArgs e)
{
    string    ConnectionString    =    "Provider=Microsoft.Jet.OLEDB.4.0;"    +    "Data
Source=Data.mdb";

    OleDbConnection connection = new OleDbConnection();
    connection.ConnectionString = ConnectionString;
    OleDbCommand myCommand;
    string sql;
    string kriteriy = comboBox1.Text.ToString();
    string znachennya = textBox6.Text.ToString();

    sql = "SELECT * From Персонал WHERE " + kriteriy + " LIKE" + " '%" + znachennya
+ "%'";

    myCommand = new OleDbCommand(sql, connection);
    connection.Open();
    OleDbDataAdapter da = new OleDbDataAdapter(myCommand);
    DataSet ds = new DataSet();
    da.Fill(ds, "Result");

```

```

        dataGridView1.DataSource = ds.Tables["Result"].DefaultView;
        connection.Close();
    }

    private void button6_Click_1(object sender, EventArgs e)
    {
        groupBox3.Visible = false;
        this.Height = 255;
        this.Width = 885;
    }

}

public partial class Chek : Form
{
    public Chek()
    {
        InitializeComponent();
    }

    private void Chek_Load(object sender, EventArgs e)
    {
        this.ChekTableAdapter.Fill(this.DataDataSet2.Chek);
    }
}

public partial class PriceList : Form
{
    public PriceList()
    {
        InitializeComponent();
    }
}

```

```
}

private void PriceList_Load(object sender, EventArgs e)
{

    this.ТовариTableAdapter.Fill(this.DataDataSet.Товари);

    this.reportViewer1.RefreshReport();
    this.reportViewer1.RefreshReport();
    this.reportViewer1.RefreshReport();
}
}
}
```

ДОДАТОК В
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька Політехніка»

Програмне забезпечення для управління магазином

Гулак Владислав Максимович
Студент групи КНТ-210СП
Федорончак Тетяна Василівна
к.т.н., доцент НУ «Запорізька Політехніка»

Рисунок В.1 – Слайд 1

- **Об'єктом** роботи є процес обліку документів та замовлень у сфері торгівлі
- **Предметом** роботи є програмне забезпечення для управління магазином
- **Мета роботи** - розробка програмного забезпечення для покращення якості роботи працівників сфери торгівлі за допомогою інструментарію для контролю та обліку товарів та іншого. Допомога в оформленні замовлень.

Рисунок В.2 – Слайд 2

Завдання до роботи

- ▶ Дослідження проблеми використання систем управління у сфері торгівлі
- ▶ Проаналізувати існуючі аналоги
- ▶ Визначити вимоги до розроблюваного програмного застосунку
- ▶ Проєктування програмного забезпечення для управління магазином
- ▶ Програмна реалізація застосунку

Рисунок В.3 – Слайд 3

Аналіз аналогів

Критерій	UiproRetail	ZStore	Розроблюваний продукт
Безкоштовний	-	+	+
Контроль за товарами	+	+	+
Вимоги до системи	-	+	+
Зручність інтерфейсу	+	++	+
Швидкість навчання роботі	-	-	+
Робота з хмарними технологіями	+	+	-

Рисунок В.4. – Слайд 4

Порівняння мов програмування

	C#	Python	Java
Швидкість роботи	++	+	+
Синтаксис	+	+	++
Підтримка .NET	+	-	-
Легкість навчання	+	++	+

Рисунок В.5 – Слайд 5

Діаграма прецедентів



Рисунок В.6 – Слайд 6

Функціонал системи

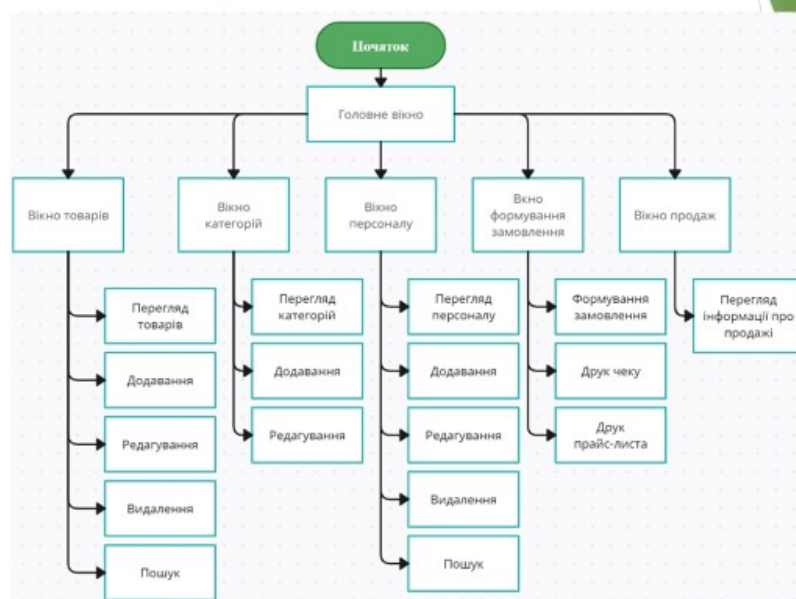


Рисунок В.7 – Слайд 7

База даних

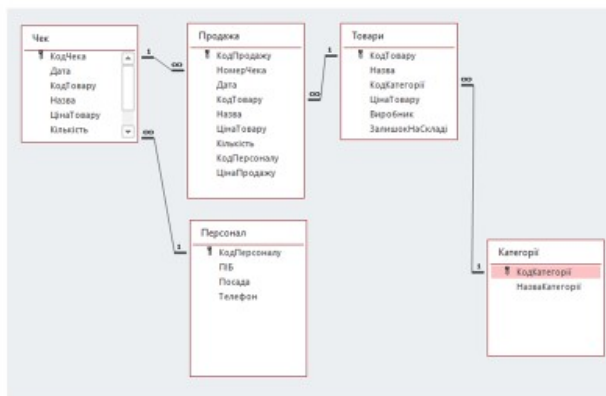


Рисунок В.8 – Слайд 8

Тестування роботи програми

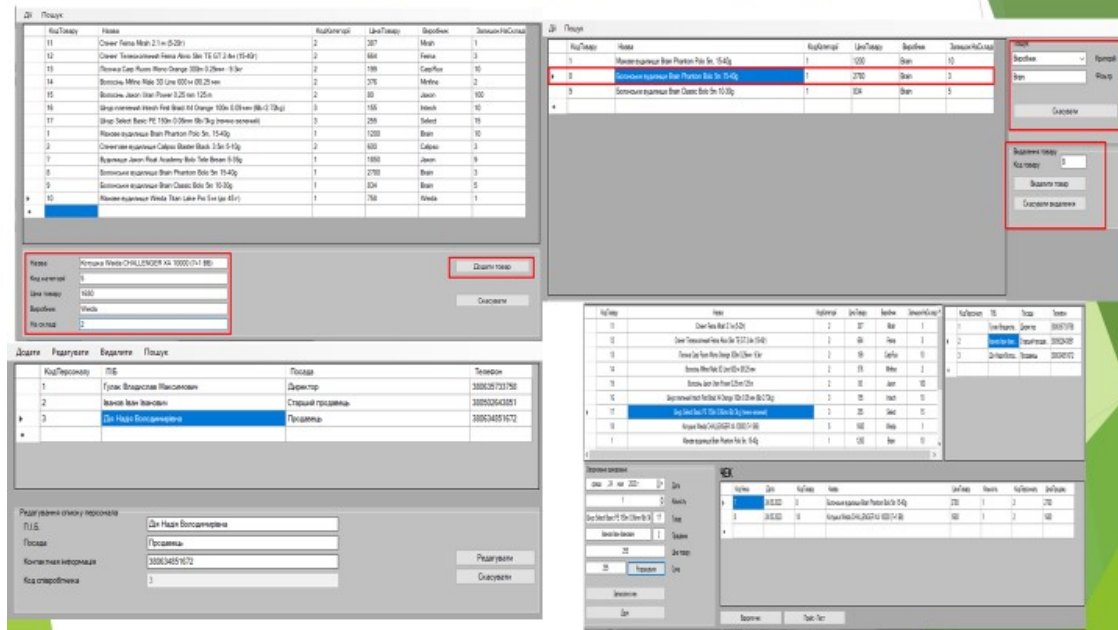


Рисунок В.9 – Слайд 9

Висновки

- Проаналізувавши проблеми та тенденції у сфері торгівлі, провівши аналіз вже існуючих аналогічних програмних рішень були сформовані вимоги до створюваного програмного продукту, його функціонал та направленість на конкретний тип підприємництва такого як мала роздрібна торгівля.
- Розроблено структуру функціоналу програмного застосунку на основі якої було виконано всі поставлені задачі. Розроблена програма має простий і зрозумілий користувацький інтерфейс, програма зручна для використання, роботі з нею легко вчитись.
- Систему простестовано за різними критеріями, в результаті чого отримано стабільну готову до роботи програму.
- Розроблений продукт реалізував всі поставлені задачі дипломної кваліфікаційної роботи бакалавра.

Рисунок В.10 – Слайд 10



Дякую за увагу

Рисунок В.11 – Слайд 11