Міністерство освіти і науки України

Національний університет водного господарства та природокористування

Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки

Кафедра комп`ютерних технологій та економічної кібернетики

**КУРСОВА РОБОТА**

з дисципліни «Програмування»

на тему:

«Інформаційна система шкільної столової»

**Виконав:** студент 2-го курсу ННІАКОТ групи ІСТ-21 спеціальності 126 «Інформаційні системи та технології»

***Призвіще Ім’я ПоБатькович***

**Науковий керівник:** ст. викладачка кафедри комп’ютерних технологій та економічної кібернетики

***Шевченко Ірина Мавіївн******а***

**Рівне – 2023**

**ЗМІСТ**

[ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 4](#_Toc152813436)

[ВСТУП 5](#_Toc152813437)

[РОЗДІЛ 1](#_Toc152813438) [ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ 7](#_Toc152813439)

[1.1 Розгорнутий опис предметної області 7](#_Toc152813440)

[1.2 Вихідні дані за варіантом завдання 8](#_Toc152813441)

[1.3 Набір функцій, які повинна вирішувати система 9](#_Toc152813442)

[1.4 Засоби проектування та програмні засоби 10](#_Toc152813443)

[РОЗДІЛ 2](#_Toc152813444) [АНАЛІЗ І ПРОЕКТУВАННЯ СИСТЕМИ 11](#_Toc152813445)

[2.1 Опис системи за об'єктно-орієнтованим підходом 11](#_Toc152813446)

[2.2 Приклад тестових вхідних та вихідних даних 12](#_Toc152813447)

[РОЗДІЛ 3](#_Toc152813448) [ОПИС ПРОГРАМИ 14](#_Toc152813449)

[3.1 Перелік модулів та функцій програмного засобу 14](#_Toc152813450)

[3.2 Діаграми 15](#_Toc152813451)

[РОЗДІЛ 4](#_Toc152813452) [КЕРІВНИЦТВО КОРИСТУВАЧА 17](#_Toc152813453)

[4.1 Вимоги до апаратного забезпечення 17](#_Toc152813454)

[4.2 Опис роботи з програмою 18](#_Toc152813455)

[ВИСНОВКИ 22](#_Toc152813456)

[СПИСОК ЛІТЕРАТУРИ 24](#_Toc152813457)

[ДОДАТОК A 26](#_Toc152813458)

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД — база даних.

СУБД (система управління базами даних) — це спеціальний пакет програм, що забезпечує створення, супроводження і використання баз даних багатьма користувачами.

CRUD (англ. create read update delete) — 4 базові функції управління даними «створення, зчитування, зміна і видалення».

ІС — інформаційна система.

# ВСТУП

У сучасному інформаційному суспільстві важливо визнати роль та значення інформаційних систем в управлінні та оптимізації різноманітних сфер життя. Однією з таких сфер є організація шкільної столової, де важливо забезпечити ефективне управління продуктами, замовленнями та клієнтською базою.

Цей курсовий проект присвячений створенню та вдосконаленню інформаційної системи для шкільної столової, спрямованої на управління продуктами. Система передбачає категоризацію кожного продукту, забезпечуючи зручний та організований процес управління широким асортиментом харчових продуктів. Важливим аспектом є використання бази даних для зберігання інформації про продукти, клієнтів та замовлення.

**Актуальність дослідження.** Інтеграція інформаційних технологій у сферу управління шкільною столовою є питанням вельми актуальним та необхідним у сучасному освітньому середовищі. Зростання обсягів замовлень, широка різноманітність продуктів та постійне прагнення підвищити рівень обслуговування ставлять під сумнів ефективність традиційних методів управління. Інформаційна система шкільної столової, яка забезпечить категоризацію продуктів та оптимізацію процесів оформлення замовлень, стане ключовим інструментом у покращенні управлінської ефективності та якості обслуговування.

**Завдання дослідження.** Розробка інформаційної системи:

Створення функціональної та зручної системи, яка забезпечить категоризацію продуктів та інтеграцію бази даних для ефективного управління клієнтами, замовленнями та запасами.

Оптимізація процесів замовлення: Визначення та реалізація ефективних процесів оформлення замовлень, з використанням інформаційної системи, з метою мінімізації часових витрат та запобігання помилкам.

Додавання та видалення інформації: Розробка механізмів додавання та видалення інформації про клієнтів, замовлення та продукти для забезпечення актуальності та коректності бази даних.

Тестування та оцінка ефективності: Проведення тестів для перевірки функціональності інформаційної системи та оцінка її впливу на ефективність управління продуктами у шкільній столовій.

**Засоби реалізації завдань проектування.** Для досягнення поставлених цілей та вирішення задач використовується об'єктно-орієнтований підхід, який дозволяє створювати моделі реальних об'єктів та їх взаємодії в межах системи управління клубом альпіністів. Реалізація проекту базується на використанні технології WPF, мови програмування C#, платформи Entity Framework для взаємодії з базою даних та інших інструментів, що відповідають сучасним стандартам розробки програмного забезпечення.

# РОЗДІЛ 1

# ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

## 1.1 Розгорнутий опис предметної області

Предметна область даного дослідження тісно пов'язана із сучасним управлінням та організацією шкільної столової. Розглядаючи широкий спектр завдань та вимог до оптимізації процесів, цей розділ розкриває ключові аспекти функціонування та особливості предметної області.

Шкільна столова є невід'ємною частиною освітнього закладу, де надається харчування для учнів та педагогічного персоналу. Це місце не лише забезпечує необхідність у відповідному харчуванні, а й відіграє важливу роль у формуванні здорових звичок серед навчальної громади.

З урахуванням різноманітності харчових продуктів та дієвого управління їхнім асортиментом, система повинна забезпечувати можливість категоризації, класифікації та зручного пошуку необхідних продуктів для готування.

Для ефективного вирішення завдань управління шкільною столовою використовується база даних, що включає інформацію про продукти, клієнтів та замовлення. Об'єднання цієї інформації в інформаційній системі дозволяє оперативно та систематично використовувати дані для оптимізації процесів.

Ключовим аспектом є процес оформлення замовлення, який передбачає точне та швидке внесення даних про продукти та клієнта в інформаційну систему. Це визначальний момент, що впливає на швидкість обслуговування та якість послуг.

Необхідність систематичного оновлення бази даних стає актуальною, тому важливим є механізм додавання та видалення інформації про клієнтів, замовлення та продукти, що дозволяє підтримувати актуальність та достовірність даних.

Розглядаючи всі аспекти предметної області, загальною метою є створення ефективної інформаційної системи для управління шкільною столовою, яка не лише задовольняє потреби учасників освітнього процесу, але й сприяє підвищенню якості обслуговування та оптимізації робочих процесів.

## 1.2 Вихідні дані за варіантом завдання

Вихідні дані для розробки інформаційної системи управління шкільною столовою включають в себе ключові аспекти, необхідні для створення ефективної та функціональної системи. Почнемо з категорій продуктів, які визначають широкий спектр доступних харчових товарів у столовій.

База даних продуктів включає інформацію про кожен продукт, визначаючи його назву, виробника, ціну, кількість на складі та термін придатності. Ці дані дозволяють системі ефективно взаємодіяти з асортиментом продуктів та підтримувати їх актуальність.

Інформація про клієнтів включає дані про основні характеристики клієнтів, такі як ім'я, контактна інформація та ідентифікаційні дані. Це становить основу для створення ідентифікаційних профілів клієнтів у системі та дозволяє зручно ведення інформації про замовлення.

Механізм оформлення замовлення передбачає створення інтерфейсу для введення даних про замовлення, включаючи вибір продуктів, їх кількість та призначення конкретному клієнтові. Це дозволяє оперативно та зручно реалізовувати замовлення та взаємодіяти з іншими частинами системи.

Засоби додавання та видалення інформації передбачають використання SQL-запитів або процедур для виконання операцій додавання та видалення записів в таблицях бази даних. Це забезпечує актуальність та консистентність інформації в системі.

Технологічні засоби для реалізації інформаційної системи включають використання платформи WPF, мови програмування C#, технології Entity Framework для роботи з базою даних та інших інструментів, що визначають сучасні стандарти розробки програмного забезпечення.

Усі вищезазначені вихідні дані формують основу для проектування та реалізації інформаційної системи, яка буде спрямована на ефективне управління продуктами та оптимізацію робочих процесів у шкільній столовій.

## 1.3 Набір функцій, які повинна вирішувати система

Розроблюючи систему для управління шкільною столовою, метою є створення функціонального інструментарію, який забезпечить ефективне взаємодію з базою даних та спростить рутинні завдання. Основна функціональність системи передбачає вирішення кількох ключових завдань.

Додавання та видалення продуктів.

Система повинна надавати можливість додавання нових продуктів до бази даних, а також вилучення зі списку існуючих. Функція додавання продуктів передбачає введення категорії, назви та ціни, щоб ідентифікувати та описати кожен продукт. За необхідності, користувач повинен мати змогу видаляти продукти, які вже не актуальні чи не використовуються.

Оформлення та видалення замовлень.

Система має підтримувати створення та видалення замовлень. Користувач повинен мати можливість оформляти замовлення, вказуючи конкретний продукт, клієнта та дату. Функція видалення замовлень дозволяє корегувати існуючі замовлення в разі необхідності.

Управління людьми (клієнтами).

Система повинна надавати функції управління клієнтами, які включають додавання нових та видалення існуючих користувачів. Додавання клієнтів включає в себе введення необхідних даних про клієнта, таких як ім'я чи контактна інформація.

Оновлення списків продуктів та клієнтів.

Система повинна забезпечувати можливість оновлення списків продуктів та клієнтів для забезпечення актуальності даних. Ця функція дозволяє користувачеві завжди мати доступ до найновішої інформації про продукти та клієнтів.

Ініціалізація та налаштування бази даних.

Система має включати ініціалізаційні функції для створення та налаштування бази даних при запуску програми. Це важливий етап для забезпечення коректної роботи системи та створення необхідних таблиць для зберігання даних.

Враховуючи ці основні функції, система буде забезпечувати повний функціонал для управління шкільною столовою, включаючи обробку продуктів, замовлень та інформації про клієнтів.

## 1.4 Засоби проектування та програмні засоби

При розробці програмного продукту для роботи клубу альпіністів засобами ООП були використані сучасні засоби проектування та програмування:

Windows Presentation Foundation (WPF): WPF був обраний для створення графічного інтерфейсу користувача через його гнучкість та можливості кастомізації елементів інтерфейсу.

Entity Framework Core: Entity Framework Core був використаний для роботи з базою даних, оскільки це потужний та зручний ORM (Object-Relational Mapping) для роботи зі структурованими даними.

C# і .NET: Мова програмування C# та платформа .NET вибрані для їхньої високої продуктивності та широких можливостей розробки.

# РОЗДІЛ 2

# АНАЛІЗ І ПРОЕКТУВАННЯ СИСТЕМИ

## 2.1 Опис системи за об'єктно-орієнтованим підходом

Система управління шкільною столовою спроектована з використанням об'єктно-орієнтованого підходу, що дозволяє створити модель, відображаючи об'єкти та їх взаємодії в межах системи управління. Основні об'єкти системи включають "Продукт", "Замовлення", "Категорія" та "Клієнт".

Клас "Продукт":

Опис: Клас представляє продукт, який може бути придбаний в шкільній столовій. Має властивості, такі як назва, виробник, ціна, категорія та термін придатності.

Методи: Клас має методи для додавання нового продукту (AddProduct) та видалення існуючого (RemoveProduct).

Клас "Замовлення":

Опис: Клас відображає замовлення, яке складається з продуктів, клієнта та дати замовлення. Містить властивості, такі як перелік продуктів, клієнта, дату та унікальний ідентифікатор.

Методи: Має методи для додавання нового замовлення (PlaceOrder) та видалення існуючого (CancelOrder).

Клас "Категорія":

Опис: Представляє категорію, в яку входять різні продукти. Має властивості, такі як назва та унікальний ідентифікатор.

Методи: Включає методи для створення нової категорії (CreateCategory) та її видалення (DeleteCategory).

Клас "Системний Контекст":

Опис: Це клас, який взаємодіє з базою даних та координує роботу інших класів. Унаслідується від контексту бази даних, включаючи DbSet для кожного основного об'єкта.

Методи: Забезпечує методи для ініціалізації бази даних (InitializeDatabase), забезпечення доступу до об'єктів та взаємодії з даними.

Застосування об'єктно-орієнтованого підходу дозволяє створити чітку модель предметної області та полегшити розробку, підтримку та розширення системи управління шкільною столовою. Кожен клас відповідає конкретній сутності, а взаємодії між ними моделюють реальний процес управління столовою.

## 2.2 Приклад тестових вхідних та вихідних даних

Для наглядного розуміння роботи системи управління шкільною столовою приведемо приклад тестових вхідних та вихідних даних для ряду типових сценаріїв використання.

Сценарій: Додавання нового продукту

Вхідні дані:

Назва продукту: "Яблука"

Виробник: "Локальний фермер"

Ціна: 25 грн

Категорія: "Фрукти"

Вихідні дані:

Успішне додавання нового продукту до бази даних.

Підтвердження додавання через інтерфейс користувача.

Сценарій: Оформлення нового замовлення

Вхідні дані:

Клієнт: Іванов Іван

Продукти: Яблука (2 шт.), Молоко (1 пакет)

Дата: 2023-12-10

Вихідні дані:

Створення нового замовлення з вказаними продуктами, клієнтом та датою.

Оновлення інтерфейсу для відображення нового замовлення у списку.

Сценарій: Видалення клієнта з системи

Вхідні дані:

Клієнт: Іванов Іван

Вихідні дані:

Видалення клієнта з бази даних.

Всі замовлення, пов'язані з цим клієнтом, також автоматично видаляються.

Сценарій: Додавання нової категорії продуктів

Вхідні дані:

Назва категорії: "Солодощі"

Вихідні дані:

Створення нової категорії у системі.

Можливість вибору цієї категорії при додаванні нових продуктів.

Ці сценарії використання тестують ключові функції системи та демонструють її здатність ефективно взаємодіяти з вхідними даними, обробляти їх та повертати очікувані результати.

# РОЗДІЛ 3

# ОПИС ПРОГРАМИ

## 3.1 Перелік модулів та функцій програмного засобу

Програмний засіб "Система управління шкільною столовою" організований за модульною структурою, яка дозволяє ефективно взаємодіяти з різними аспектами функціоналу. Нижче наведено перелік модулів та їх основних функцій.

Модуль OrdersModule містить функції для роботи з замовленнями. Метод SelectOrders використовує контекст бази даних, щоб отримати дані про замовлення та відобразити їх у вказаному DataGrid. Метод AddOrder додає нове замовлення із вказаним продуктом, клієнтом та датою, використовуючи контекст бази даних. Метод DelOrder видаляє замовлення за його унікальним ідентифікатором.

Модуль ProductsModule включає функції для роботи з продуктами. Метод SelectFood використовує контекст бази даних для отримання та відображення даних про продукти у вказаному DataGrid. Метод AddFood додає новий продукт із вказаною назвою, категорією та ціною, використовуючи контекст бази даних. Метод DelFood видаляє продукт за його унікальним ідентифікатором.

Модуль ClientsModule містить функції для роботи з клієнтами. Метод SelectPeople використовує контекст бази даних для отримання та відображення даних про клієнтів у вказаному DataGrid. Метод AddPeople додає нового клієнта із вказаною повною назвою, використовуючи контекст бази даних. Метод DelPeople видаляє клієнта за його унікальним ідентифікатором.

Модуль ComboBoxModule містить функції для роботи з ComboBox. Методи SelectComboBoxPeople та SelectComboBoxFood заповнюють відповідні ComboBox інформацією про клієнтів та продукти, використовуючи контекст бази даних. Кожен з цих модулів відповідає за конкретний аспект функціоналу системи, забезпечуючи виконання відповідних операцій.

Кожен з цих модулів відповідає за конкретний аспект функціоналу системи та має визначені функції для ефективного виконання відповідних операцій.

## 3.2 Діаграми

Діаграми програми, що розробляється, зображено на рисунках 3.1 та 3.2 відповідно.

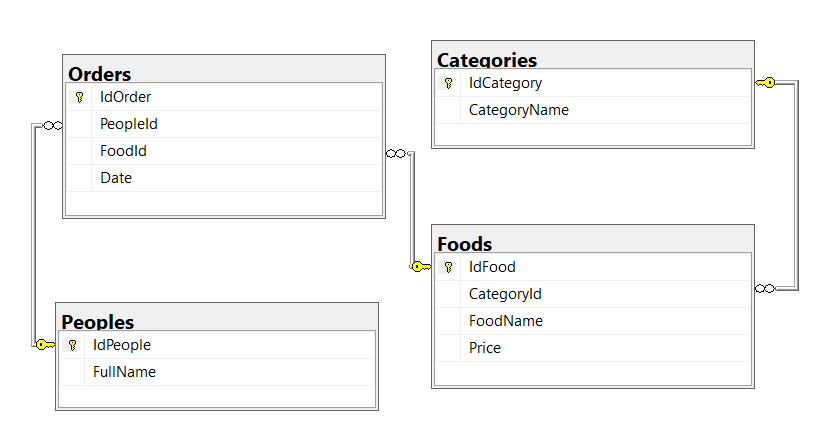


Рисунок 3.1 – Діаграма бази даних

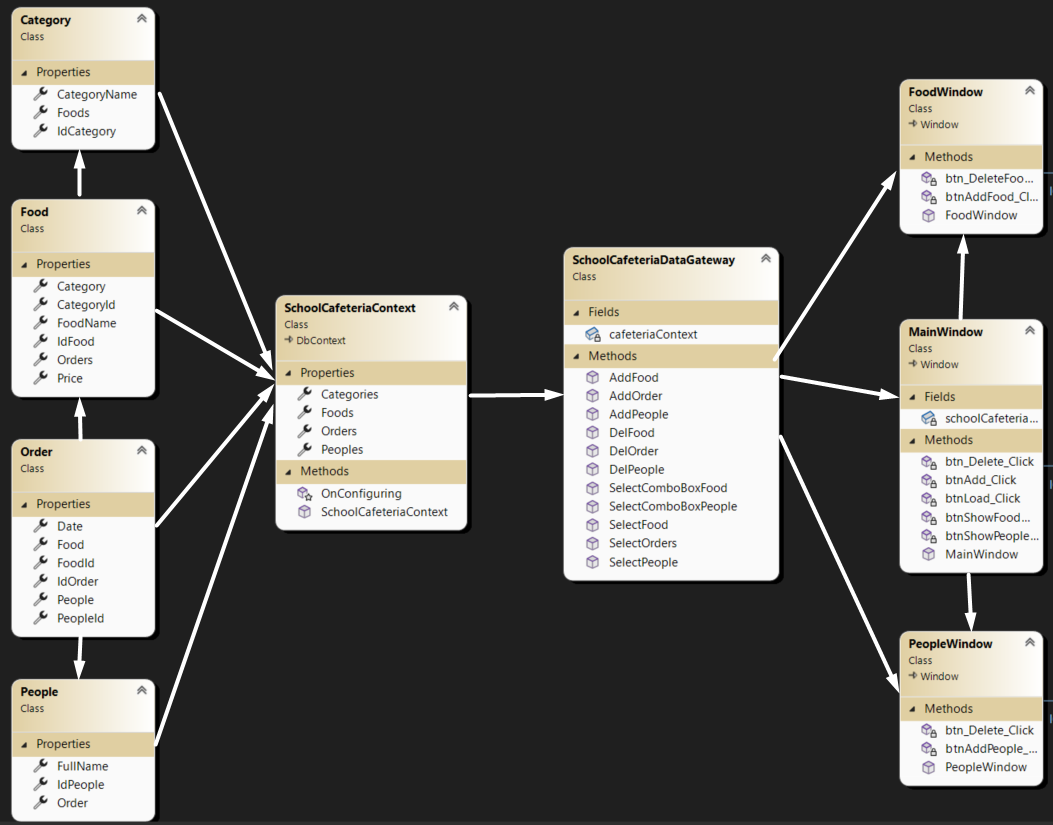


Рисунок 3.2 – Діаграма класів

# РОЗДІЛ 4

# КЕРІВНИЦТВО КОРИСТУВАЧА

## Вимоги до апаратного забезпечення

Для ефективної роботи програмного засобу "Система управління шкільною столовою" необхідно враховувати певні вимоги до апаратного забезпечення. Мінімальні та рекомендовані вимоги визначаються з урахуванням оптимальності роботи додатку та задоволення потреб користувачів.

Мінімальні вимоги:

Процесор: Двоядерний процесор з тактовою частотою не менше 2 ГГц.

Оперативна пам'ять: Мінімум 4 ГБ RAM.

Дисковий простір: Наявність не менше 20 ГБ вільного місця на жорсткому диску.

Відеокарта: Графічний адаптер з підтримкою DirectX 11.

Рекомендовані вимоги:

Процесор: Чотириядерний або більше процесор з тактовою частотою 3 ГГц або вище.

Оперативна пам'ять: Рекомендовано не менше 8 ГБ RAM для оптимальної продуктивності.

Дисковий простір: SSD з обсягом не менше 50 ГБ для швидкого доступу до даних.

Відеокарта: Високоефективний графічний адаптер із підтримкою DirectX 12 для забезпечення високоякісного відображення графіки.

Важливо також зазначити, що реальні вимоги можуть змінюватися в залежності від обсягу та характеру роботи користувачів, а також від можливостей розвитку програмного забезпечення у майбутньому.

## Опис роботи з програмою

Система управління шкільною столовою є високофункціональним програмним засобом, спроектованим для ефективного вирішення завдань управління продуктами, клієнтами та замовленнями. Нижче подано докладний опис основних функцій та процесів роботи з програмою.

При запуску попадаємо у головну вікно програми, можемо побачити кнопки «Управління постачальниками», «Управління продуктами» та вкладку «Замовлення», Рис. 4.1.

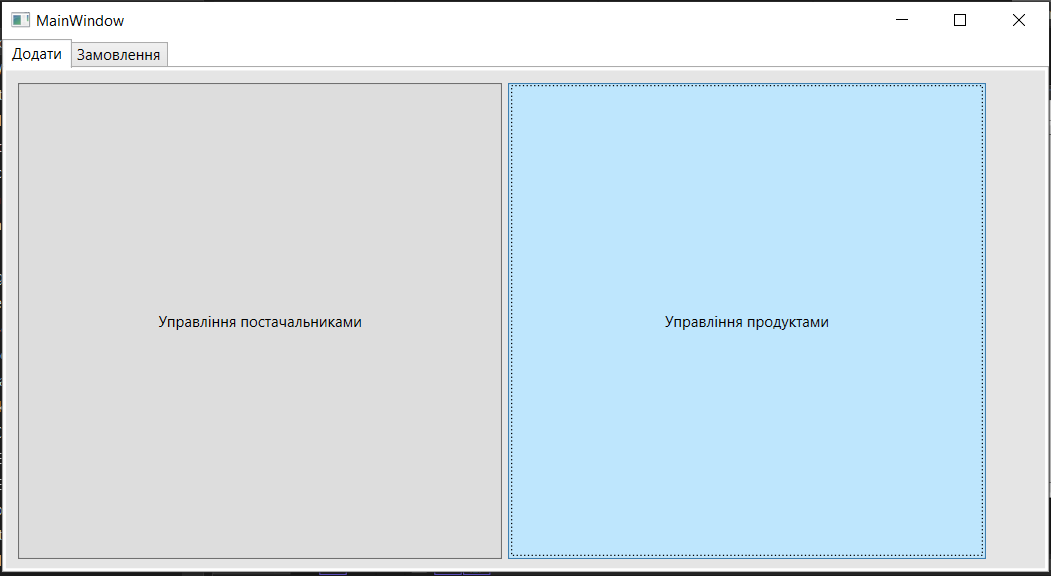


Рисунок 4.1 – Головне вікно програми

У вікні «Управління постачальниками» можемо переглянути, додати чи видалити постачальників, Рис. 4.2.

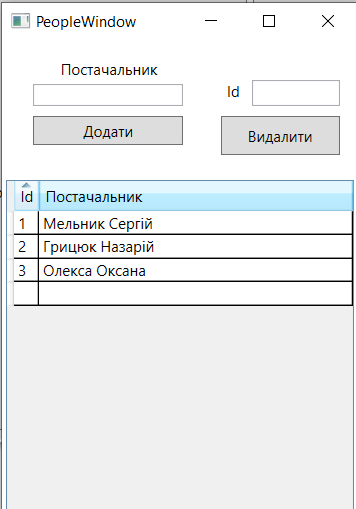


Рисунок 4.2 – Управління постачальниками

По аналогії і з управлінням продуктами, Рис. 4.3.

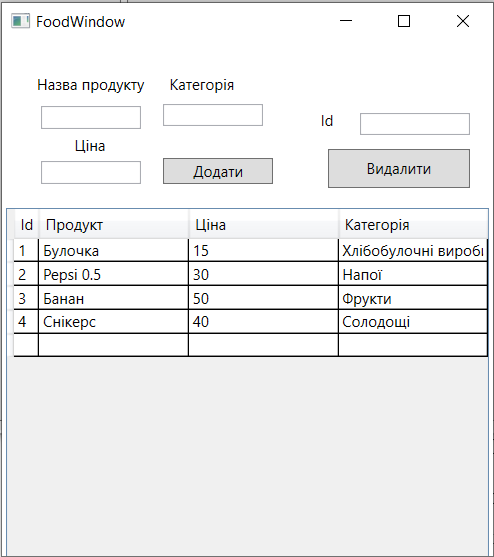


Рисунок 4.3 – Управління продуктами

На вкладці «Замовлення» можемо переглянути, додати, видалити та оновити замовлення, Рис. 4.4.

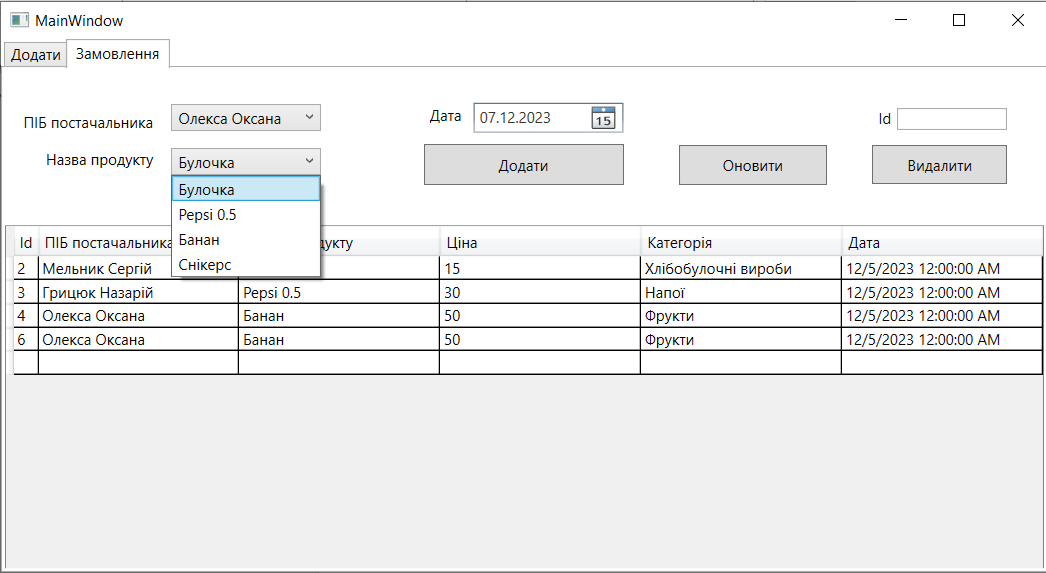


Рисунок 4.4 – Замовлення

Також в таблиці можемо сортувати дані за будь яким параметром, Рис. 4.5.

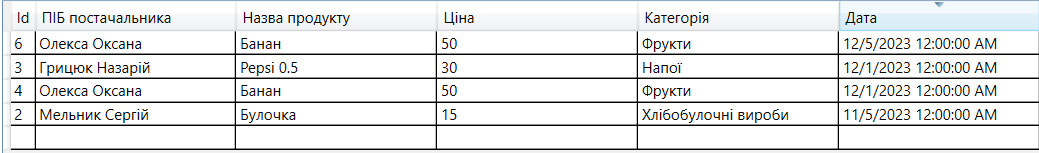


Рисунок 4.5 – Сортування

# ВИСНОВКИ

У ході дослідження та розробки інформаційної системи для управління шкільною столовою було проведено кілька ключових етапів, визначаючи обсяг робіт та впроваджуючи засоби для досягнення поставлених цілей.

Аналіз потреб та визначення завдань:

- Проведено докладний аналіз функціональних та управлінських потреб шкільної столової.

- Визначено основні завдання для розробки інформаційної системи, зокрема категоризація продуктів, оптимізація замовлень та управління базою даних.

Розробка інформаційної системи:

- Застосовано об'єктно-орієнтований підхід для створення моделей реальних об'єктів та їх взаємодії.

- Використано технологію WPF для розробки графічного інтерфейсу користувача, що забезпечує зручність та естетичний дизайн.

- Мова програмування C# використовувалася для програмної логіки та взаємодії з базою даних.

- Для роботи з базою даних використано платформу Entity Framework.

Реалізація функцій системи:

- Розроблено механізми категоризації продуктів та їх інтеграції в систему управління клієнтами та замовленнями.

- Оптимізовано процеси замовлення для мінімізації часових витрат та уникнення помилок.

- Додано та видалено інформацію про клієнтів, замовлення та продукти для забезпечення актуальності бази даних.

Загалом, впровадження інформаційної системи було проведено з використанням передових технологій та здійснено спільними зусиллями команди, що забезпечило створення функціонального та ефективного засобу для покращення управлінських процесів у шкільній столовій.

# СПИСОК ЛІТЕРАТУРИ

1. Обґрунтування вибору мови програмування [Електронний ресурс] URL:<https://studopedia.ru/15_59823_obruntuvannya-viboru-movi-programuvannya.html>
2. Об’єктно-орієнтоване програмування. Частина 1. Основи об’єктноорієнтованого програмування на мові C#.: Навчальний посібник. / Д.В. Настенко, А. Б. Нестерко. – К.: НТУУ «КПІ», 2016. - 76с. URL: <https://ela.kpi.ua/bitstream/123456789/16671/1/OOP_manual.pdf>
3. Обзор функционала SQL Server Management Studio (SSMS) [Електронний ресурс] URL:<https://info-comp.ru/ssms-feature-overview#chto-takoe-sql-server-management-studio>
4. Структура мови SQL Structured Query Language [Електронний ресурс] URL: <https://ua-referat.com/Структура_мови_SQL_Structured_Query_Language>
5. Кібернетика та інформаційні системи [Електронний ресурс] URL: <https://www.bestreferat.ru/referat-301142.html>
6. Проектування, розробка та впровадження БД ІС в економічну діяльність підприємства (на прикладі ДП "Алушталіфт") [Електронний ресурс] URL: <http://ukrefs.com.ua/print:page,1,171932-Proektirovanie-razrabotka-i-vnedrenie-BD-IS-v-ekonomicheskuyu-deyatel-nost-predpriyatiya-na-primere-GP-Alushtalift.html>
7. Інформаційні системи і технології в фінансових установах [Електронний ресурс] URL:<http://5fan.ru/wievjob.php?id=36099>
8. Інформаційні системи з базами даних [Електронний ресурс] <URL:https://elearning.sumdu.edu.ua/free_content/lectured:89b3d175c06a6b137e410cb14821d0e94549ad5a/latest/44150/index.html>
9. Поняття бази даних, системи управління базами даних (СУБД) [Електронний ресурс] URL:<https://www.sites.google.com/site/lutskschool1yasenchuk/materiali-do-urokiv/11-klas/urok-17>
10. Каштан В.Ю., Іванов Д.В. Конспект лекцій з дисципліни “Бази даних в інформаційних системах”. Для студентів галузі знань 12 “Інформаційні технології” спеціальності 126 “Інформаційні системи та технології”. – Д.: НТУ «ДП», 2020. – 58 с. URL:<https://it.nmu.org.ua/ua/scientific_method_materials/lecture_notes/Конспект_лекцій_БД_частина1_2020.pdf>
11. Організація баз даних та знань [Електронний ресурс] URL: <https://www.bestreferat.ru/referat-377919.html>
12. Руководство по Entity Framework Core [Електронний ресурс] URL: <https://metanit.com/sharp/entityframeworkcore>
13. YouTube. (2019, січень 22). SQL базы данных и Entity Framework в C – Учим Шарп #19 [Відеофайл] URL:<https://www.youtube.com/watch?v=3yb4idCg-Qs>
14. YouTube. (2017, жовтень 21). WPF Entity Framewok CRUD Create / Insert Read Update Delete [Відеофайл] <URL:https://youtu.be/VGRvi4-1VhA>

# ДОДАТОК A

Лістинг програми (MainWindow)

public partial class MainWindow : Window

{

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway;

public MainWindow()

{

try

{

InitializeComponent();

schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.SelectOrders(dataGridOrder);

schoolCafeteriaDataGateway.SelectComboBoxFood(comboBoxFood);

schoolCafeteriaDataGateway.SelectComboBoxPeople(comboBoxPeople);

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.AddOrder(comboBoxFood.Text, comboBoxPeople.Text, DateTime.Parse(datePickerDate.Text));

schoolCafeteriaDataGateway.SelectOrders(dataGridOrder);

}

private void btnShowPeopleWindow\_Click(object sender, RoutedEventArgs e)

{

PeopleWindow peopleWindow = new PeopleWindow();

peopleWindow.Show();

}

private void btnShowFoodWindow\_Click(object sender, RoutedEventArgs e)

{

FoodWindow foodWindow = new FoodWindow();

foodWindow.Show();

}

private void btn\_Delete\_Click(object sender, RoutedEventArgs e)

{

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.DelOrder(int.Parse(tb\_Del.Text));

schoolCafeteriaDataGateway.SelectOrders(dataGridOrder);

}

}

Лістинг програми (FoodWindow)

public partial class FoodWindow : Window

{

public FoodWindow()

{

InitializeComponent();

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.SelectFood(dataGridFood);

}

private void btnAddFood\_Click(object sender, RoutedEventArgs e)

{

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.AddFood(tbCategory.Text, tbProductName.Text, float.Parse(tbPrice.Text));

schoolCafeteriaDataGateway.SelectFood(dataGridFood);

}

private void btn\_DeleteFood\_Click(object sender, RoutedEventArgs e)

{

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.DelFood(int.Parse(tb\_DelProduct.Text));

schoolCafeteriaDataGateway.SelectFood(dataGridFood);

}

}

Лістинг програми (PeopleWindow)

public partial class PeopleWindow : Window

{

public PeopleWindow()

{

InitializeComponent();

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.SelectPeople(dataGrid);

}

private void btnAddPeople\_Click(object sender, RoutedEventArgs e)

{

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.AddPeople(tbCategoryName.Text);

schoolCafeteriaDataGateway.SelectPeople(dataGrid);

}

private void btn\_Delete\_Click(object sender, RoutedEventArgs e)

{

SchoolCafeteriaDataGateway schoolCafeteriaDataGateway = new SchoolCafeteriaDataGateway();

schoolCafeteriaDataGateway.DelPeople(int.Parse(tb\_Del.Text));

schoolCafeteriaDataGateway.SelectPeople(dataGrid);

}

}

Лістинг програми (SchoolCafeteriaDataGateway)

public class SchoolCafeteriaDataGateway

{

SchoolCafeteriaContext cafeteriaContext;

public void SelectOrders(DataGrid dataGrid)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

cafeteriaContext.Peoples.Load();

cafeteriaContext.Categories.Load();

cafeteriaContext.Foods.Load();

cafeteriaContext.Orders.Load();

dataGrid.ItemsSource = cafeteriaContext.Orders.Local.ToBindingList();

}

}

public void AddOrder(string food, string people, DateTime dateTime)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

var foodId = cafeteriaContext.Foods.Include(x=>x.Category).FirstOrDefault(x => x.FoodName == food).IdFood;

var peopleId = cafeteriaContext.Peoples.FirstOrDefault(x => x.FullName == people).IdPeople;

Order order = new Order

{

FoodId = foodId,

PeopleId = peopleId,

Date = dateTime

};

cafeteriaContext.Orders.Add(order);

cafeteriaContext.SaveChanges();

}

}

public void DelOrder(int id)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

Order del = new Order() { IdOrder = id };

cafeteriaContext.Orders.Attach(del);

cafeteriaContext.Orders.Remove(del);

cafeteriaContext.SaveChanges();

}

}

public void SelectFood(DataGrid dataGrid)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

cafeteriaContext.Peoples.Load();

cafeteriaContext.Categories.Load();

cafeteriaContext.Foods.Load();

cafeteriaContext.Orders.Load();

dataGrid.ItemsSource = cafeteriaContext.Foods.Local.ToBindingList();

}

}

public void SelectPeople(DataGrid dataGrid)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

cafeteriaContext.Peoples.Load();

cafeteriaContext.Categories.Load();

cafeteriaContext.Foods.Load();

cafeteriaContext.Orders.Load();

dataGrid.ItemsSource = cafeteriaContext.Peoples.Local.ToBindingList();

}

}

public void AddFood(string categoryName, string foodName, float price)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

Category category = new Category

{

CategoryName = categoryName

};

cafeteriaContext.Categories.Add(category);

cafeteriaContext.SaveChanges();

Food food = new Food

{

FoodName = foodName,

Price = price,

CategoryId = category.IdCategory

};

cafeteriaContext.Foods.Add(food);

cafeteriaContext.SaveChanges();

}

}

public void DelFood(int id)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

Food del = new Food() { IdFood = id };

cafeteriaContext.Foods.Attach(del);

cafeteriaContext.Foods.Remove(del);

cafeteriaContext.SaveChanges();

}

}

public void AddPeople(string fullName)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

People people = new People

{

FullName = fullName

};

cafeteriaContext.Peoples.Add(people);

cafeteriaContext.SaveChanges();

}

}

public void DelPeople(int id)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

People del = new People() { IdPeople = id };

cafeteriaContext.Peoples.Attach(del);

cafeteriaContext.Peoples.Remove(del);

cafeteriaContext.SaveChanges();

}

}

public void SelectComboBoxPeople(ComboBox comboBox)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

comboBox.ItemsSource = cafeteriaContext.Peoples.ToList();

comboBox.SelectedValuePath = "Idpeople";

comboBox.DisplayMemberPath = "FullName";

}

}

public void SelectComboBoxFood(ComboBox comboBox)

{

using (cafeteriaContext = new SchoolCafeteriaContext())

{

comboBox.ItemsSource = cafeteriaContext.Foods.ToList();

comboBox.SelectedValuePath = "IdFood";

comboBox.DisplayMemberPath = "FoodName";

}

}

}

Лістинг програми (SchoolCafeteriaContext)

public class SchoolCafeteriaContext : DbContext

{

public DbSet<People> Peoples { get; set; }

public DbSet<Category> Categories { get; set; }

public DbSet<Food> Foods { get; set; }

public DbSet<Order> Orders { get; set; }

public SchoolCafeteriaContext()

{

try

{

//Database.EnsureDeleted();

Database.EnsureCreated();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer("Server=(localdb)\\mssqllocaldb;Database=SchoolCafeteriaDB;Trusted\_Connection=True;");

}

}Лістинг програми (Order)

public class Order

{

[Key]

public int IdOrder { get; set; }

public int IdClient { get; set; }

[ForeignKey("IdClient")]

public Client Client { get; set; }

public int IdProduct { get; set; }

[ForeignKey("IdProduct")]

public Product Product { get; set; }

public DateTime DateTime { get; set; }

}

Лістинг програми (Category)

public class Category

{

[Key]

public int IdCategory { get; set; }

public string CategoryName { get; set; }

public virtual ICollection<Food> Foods { get; set; }

}

Лістинг програми (Food)

public class Food

{

[Key]

public int IdFood { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; }

public string FoodName { get; set; }

public float Price { get; set; }

public virtual ICollection<Order> Orders { get; set; }

}

Лістинг програми (Order)

public class Order

{

[Key]

public int IdOrder { get; set; }

public int PeopleId { get; set; }

public People People { get; set; }

public int FoodId { get; set; }

public Food Food { get; set; }

public DateTime Date { get; set; }

}

Лістинг програми (People)

public class People

{

[Key]

public int IdPeople { get; set; }

public string FullName { get; set; }

public virtual ICollection<Order> Order { get; set; }

}