

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь бакалавр
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інформаційних технологій, штучного
інтелекту і кібербезпеки
Сергій ГРИБКОВ

“ 15 ” квітня 2024 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Баєва Андрія Андрійовича

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи для підтримки управління діяльністю кафе
керівник роботи Литвинов Валерій Андроникович, професор, доктор технічних наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 15 квітня 2024 року № 279-кв

2. Строк подання здобувачем роботи 03.06.2024 р.

3. Вихідні дані до роботи

Структура підприємства, асортимент кафе

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) 1 Системний аналіз об'єкта автоматизації, 2 Технічне завдання, 3 Розробка комплексу задач автоматизації, 4 Охорона праці, Висновки, Список використаних джерел, Додатки

5. Перелік графічного матеріалу

Організаційна структура кафе, логічно-фізична схема моделі бази даних, діаграма бази даних, форми інформаційної системи

АНОТАЦІЯ

Дипломна робота «Розроблення інформаційної системи для підтримки управління діяльністю кафе» розроблена Баєвом А.А. складається з 97 сторінок, 18 таблиць, 55 рисунків, 4 додатків, 14 джерел.

У бакалаврській роботі наведені результати дослідження діяльності роботи кафе, обґрунтовано необхідність розробки інформаційно-облікової системи. Для реалізації функцій, описаних у функціональній моделі, здійснено вибір програмного забезпечення, наведено алгоритми реалізації функцій системи, розроблена інструкція користувача. Проведено техніко-економічне обґрунтування впровадження системи, описані заходи з охорони праці для робочого місця на підприємстві.

КЛЮЧОВІ СЛОВА: КАФЕ, MS VISUAL STUDIO 2022, КЛІЄНТ-СЕРВЕРНІ ТЕХНОЛОГІЇ, C#, ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ, ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМНИЙ ДОДАТОК, ПЗ.

ANNOTATION

The diploma thesis "Development of an Information System to Support the Management of a Café" by A.A. Baiev consists of 97 pages, 18 tables, 55 figures, 4 appendixs, and 14 sources.

The bachelor's thesis presents the results of a study on the café's operations, substantiates the need for the development of an information accounting system. To implement the functions described in the functional model, software was selected, algorithms for implementing the system's functions were provided, and a user manual was developed. A technical and economic feasibility study of the system's implementation was conducted, and occupational safety measures for the workplace at the enterprise were described.

KEYWORDS: CAFÉ, MS VISUAL STUDIO 2022, CLIENT-SERVER TECHNOLOGIES, C#, ECONOMIC EFFICIENCY, INFORMATION SYSTEM, SOFTWARE APPLICATION, SOFTWARE.

ЗМІСТ

Вступ.....	7
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ’ЄКТА АВТОМАТИЗАЦІЇ	9
1.1. Загальна характеристика	9
1.2. Організаційна структура кафе, роль і взаємодія підрозділів.....	10
1.3. Розроблення функціональної моделі.....	11
1.4. Аналіз існуючих бізнес-процесів	17
1.5. Порівняння існуючих бізнес-процесів.....	21
1.6. Розрахунок економічного ефекту від впровадження системи	32
1.7. Висновок розділу 1.....	35
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ.....	36
2.1. Загальні положення.....	36
2.2. Призначення і цілі створення системи.....	36
2.3. Характеристика об’єкта автоматизації	37
2.4. Вимоги до системи.....	37
2.5. Склад і зміст робіт по створенню системи	47
2.6. Порядок контролю і приймання системи	47
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію	48
2.8. Вимоги до документації	48
2.9. Джерела розробки	48
РОЗДІЛ 3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	49
3.1. Інформаційне забезпечення системи.....	49
3.2. Алгоритмізація та реалізація комплексу задач автоматизації.....	54
3.3. Інструкція користувача.....	73
3.4. Технічне та системне забезпечення розробки.....	86
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	91
ВИСНОВОК.....	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	96
ДОДАТКИ.....	98

ВСТУП

У сучасному світі успіх будь-якого підприємства значною мірою залежить від ефективності використання інформаційних технологій. Кафе, як і багато інших бізнесів, стикаються з необхідністю оптимізації своїх процесів, підвищення якості обслуговування клієнтів та покращення загальної ефективності. Розроблення інформаційної системи для підтримки управління діяльністю кафе може стати вирішальним кроком на шляху до досягнення цих цілей.

Основна теза цього дипломного дослідження полягає в тому, що інтеграція сучасної інформаційної системи здатна значно покращити роботу кафе. Зокрема, системи, що забезпечують автоматизацію обліку замовлень, та управління персоналом, можуть допомогти зменшити витрати часу, уникнути помилок та покращити якість послуг.

На практиці, впровадження інформаційної системи має цілу низку аспектів, які необхідно врахувати. Наприклад, необхідно забезпечити безперебійну взаємодію системи з існуючими процесами кафе. Також важливо передбачити зручність використання системи для персоналу, а також її гнучкість і масштабованість для подальшого розвитку бізнесу. Крім того, слід врахувати безпеку даних та їх захист від несанкціонованого доступу.

Основний аргумент для впровадження інформаційної системи полягає в тому, що вона може сприяти підвищенню задоволеності клієнтів, оскільки дозволяє швидше обробляти замовлення, забезпечує точність в обліку та дозволяє більш ефективно управляти кафе. Крім того, такі системи дозволяють зберігати та аналізувати дані про клієнтів, що може бути корисним для маркетингових цілей.

Проте існують і певні обмеження та проблеми, пов'язані з впровадженням інформаційних систем. Це може включати високі початкові витрати, складнощі в навчанні персоналу, а також можливість технічних збоїв. У процесі написання цього дипломного дослідження буде розглянуто, як ці

проблеми можна вирішити або мінімізувати, і які інноваційні підходи можуть бути використані для розробки ефективної інформаційної системи для кафе.

1 СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА АВТОМАТИЗАЦІЇ

1.1 Загальна характеристика

Міське середовище - це особлива територія, яка включає в себе площі, проспекти, культові споруди та інші значущі об'єкти. Давні таверни в Римі, середньовічні підвали, а потім кафе, чайні, шинки та паби – все це місця, де люди збиралися, щоб поспілкуватися і відпочити. В таких закладах завжди панувала особлива атмосфера, яку доповнювали внутрішні "кодекси" або неписані правила, що відрізняли їх один від одного. Постійні відвідувачі часто обирали той чи інший заклад через його зручне розташування чи тематичну спрямованість. Наприклад, одні паби були популярними через близькість до житла, а інші приваблювали артистів або письменників.

Спеціалізація міських закладів була чіткою і конкретною. У пабах активно обговорювали політику чи спорт, у кафе часто народжувалися літературні шедеври 19-го і 20-го століть. Кафе "Зелений Дракон", наприклад, була місцем підготовки "Бостонського чаювання", а Йоганн Себастьян Бах написав свою "Кавову кантату" в кафе Циммермана, оскільки господар вважав, що музика допоможе змінити ставлення до продукції в суспільстві.

Зміни в містобудуванні часто призводили до появи нових осередків міського життя. Так, паризька "османізація" сприяла поширенню кафе та відкритих вуличних кафе, що дозволяли людям відчувати себе частиною міста, насолоджуючись стравами та напоями і спостерігаючи за життям навколо. Цей новий формат став своєрідною реакцією на швидкий темп життя, спричинений технологічною революцією.

Громадські місця також змінювалися під впливом функціоналістських ідей, і до початку 20-го століття неформальні клуби стали частиною міської структури, відіграючи важливу роль в міському житті. Будинки культури, Палаці Будівельників та Металургів, які будувалися великими виробничими підприємствами, мали на меті зміцнення почуття спільності серед працівників. Однак, камерність і інтимність поступово зникли з міського життя. Замість

традиційних чайних з'явилися пункти швидкого харчування, де атмосфера поступилася місцем швидкості обслуговування.

Таким чином, почали з'являтися перші кафе в різних куточках світу. Вони слугували місцем, де клієнт обмінював певну суму грошей на готову продукцію. Перш ніж приступити до виконання дипломної роботи, я дослідив діяльність типової кафе[1].

1.2. Організаційна структура кафе, роль і взаємодія підрозділів

У кафе більшість ключових функцій виконує працівник, включаючи прийом замовлень та безпосередньо продаж продукції клієнтам. Організаційні питання вирішує власник, який також займається загальним керівництвом, забезпечує наявність необхідних матеріалів і стежить за облаштуванням приміщення, технічний персонал слідкує за правильною роботою автоматизованих систем та всієї іншої електроніки. На рисунку 1.1 зображена Загальна схема організаційної структури підприємства(Рис 1.1).

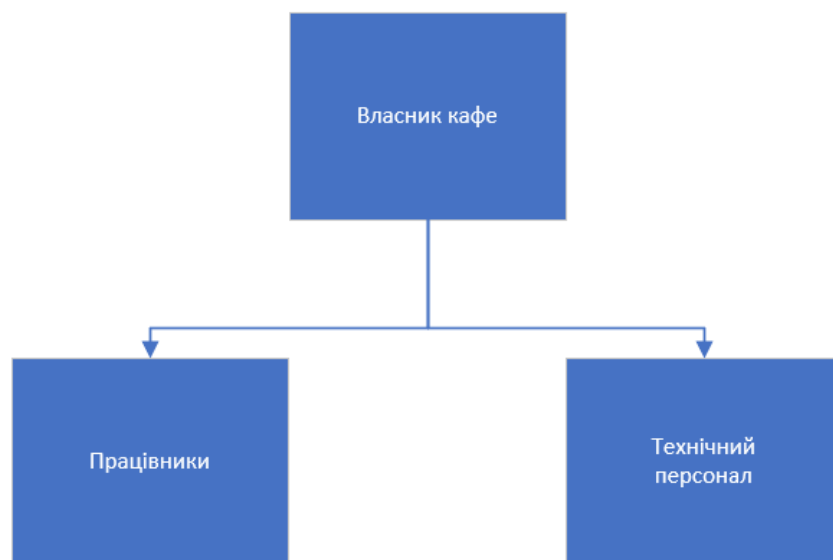


Рисунок 1.1 – Загальна схема організаційної структури підприємства

У процесі діяльності кафе виникають такі питання:

- як найкраще організувати роботу кафе;

- як ефективно вести облік у кафе.
- як організувати розклад для персоналу.

1.3. Розроблення функціональної моделі

Методологія SADT (Structured Analysis and Design Technique) дозволяє побудувати функціональну модель діяльності кафе, використовуючи серію діаграм із супровідною документацією. Основною ідеєю SADT є розбиття складної системи на менші компоненти, які потім можна докладно проаналізувати. Кожен з цих компонентів представляється у вигляді блоку на діаграмах, де показуються їхні взаємозв'язки та взаємодія із зовнішнім середовищем.

Побудова SADT-моделі починається з представлення всієї системи у вигляді одного блоку, який символізує все кафе як єдине ціле. У цьому блоці вказують загальне ім'я, яке описує всю систему. Дуги, що виходять і входять до цього блоку, представляють вхідні та вихідні зв'язки із зовнішнім середовищем, що дає уявлення про загальний контекст системи.

На наступному етапі кожен з підфункцій вихідної функції моделюється за допомогою додаткових блоків, з'єднаних інтерфейсними дугами. Кожен блок у такій декомпозиції представляє певну підфункцію, яка входить до загальної функції системи. Такий підхід забезпечує структурований і деталізований опис функцій системи, що дозволяє виділити та проаналізувати всі необхідні елементи.

Важливо, що SADT-модель надає повний опис системи, не пропускаючи жодного елемента, який був частиною вихідної функції. Кожна підфункція може бути додатково декомпозована, що дозволяє деталізувати її структуру та взаємозв'язки. Такий підхід сприяє виявленню зайвих операцій, підвищенню ефективності та аналізу вартості функцій.

1.3.1. Функціональна модель

На першому етапі моделювання створюється контекстна діаграма, яка визначає межі моделі, цілі та точку зору на систему. Головною метою функціональної моделі кафе є виявлення надмірних операцій, проведення попередніх розрахунків ефективності, а також визначення функцій, які можуть бути автоматизовані. Така модель дозволяє визначити інформаційні потоки, які є суттєвими в діяльності підприємства, та розробити рекомендації щодо оптимізації та автоматизації процесів[2].

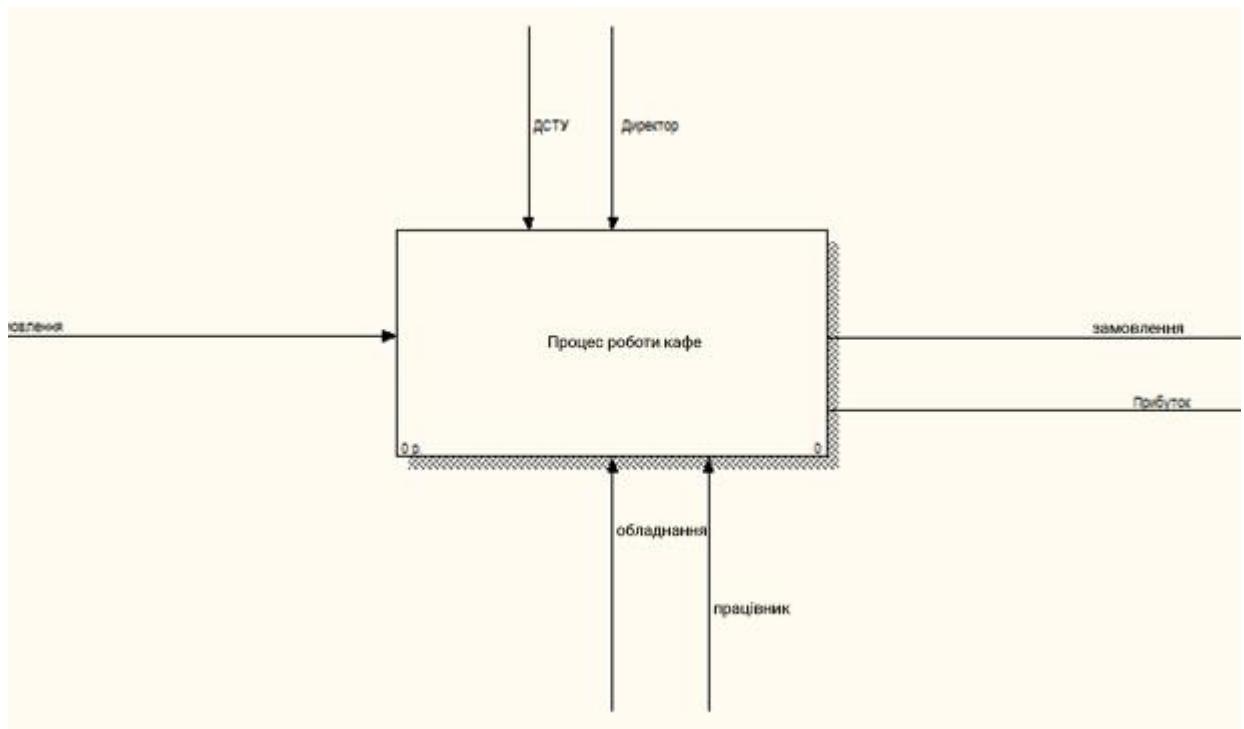


Рисунок 1.2 – Контекстна діаграма кафе

Контекстна діаграма (Рис 1.2) є важливим інструментом для відображення загального уявлення про систему і взаємодії її з оточенням. Вона зображується у вигляді прямокутника, на кожній стороні якого показані вхідні та вихідні величини, контролюючі механізми та механізми, які відповідають за ресурси. Це створює повну картину взаємодії системи із зовнішнім середовищем, дозволяючи зрозуміти основні функції, що виконуються всередині системи.

У контекстній діаграмі кафе, яка використовується для розробки інформаційної системи, показані різні аспекти роботи кафе. Вхідними величинами є клієнти, які заходять до кафе, щоб замовити продукцію. Керуючі

входи включають в себе процедури та стандарти, які визначають приготування замовлень, такі як рецептури та стандарти ДСТУ. Вихідні величини – це приготовані замовлення, що пропонуються клієнтам. Механізми, які забезпечують виконання функцій, включають комп'ютерну техніку, програмне забезпечення та обладнання для приготування продукції.

Контекстна діаграма розбивається на підфункції, які далі розглядаються більш детально за допомогою функціональної декомпозиції. При цьому кожна підфункція може бути розділена на менші компоненти, що дозволяє отримати детальний опис системи. У контексті кафе функціональна декомпозиція призводить до виділення основних функцій:

- *управління кафе*: включає управління операціями, ціноутворення, керівництво персоналом та контроль за якістю;
- *приготування продукції*: включає прийом замовлень, приготування продукції за допомогою відповідного обладнання, видачу готового продукту клієнтам;
- *управління рахунками*: включає облік продажів, прийняття грошей, ведення бухгалтерії та фінансових звітів;

В рамках приготування продукції виділяються наступні підфункції:

- *прийом замовлень*: працівник приймає замовлення від клієнтів та вводить його в систему;
- *приготування продукції*: процес приготування продукції відповідно до рецептури;
- *подача продукції*: передача приготованої продукції клієнтам;
- *прийняття грошей*: облік фінансових транзакцій та видача здачі клієнтам;

Ця функціональна декомпозиція дозволяє чітко розмежувати різні аспекти діяльності кафе та виявити області, які можуть бути оптимізовані або автоматизовані. Це допомагає створити ефективну інформаційну систему для підтримки діяльності кафе.

Діаграма декомпозиції зображені на рисунку 1.3 та на рисунку 1.4.

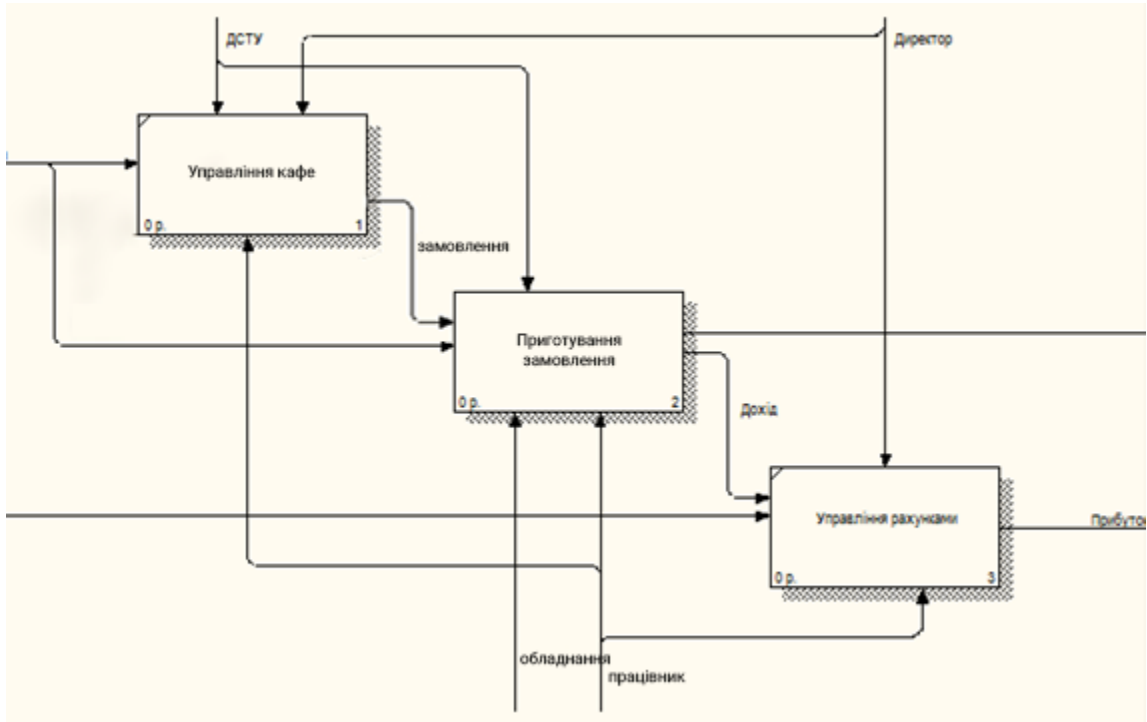


Рисунок 1.3 – Діаграма декомпозиції контекстної діаграми кафе

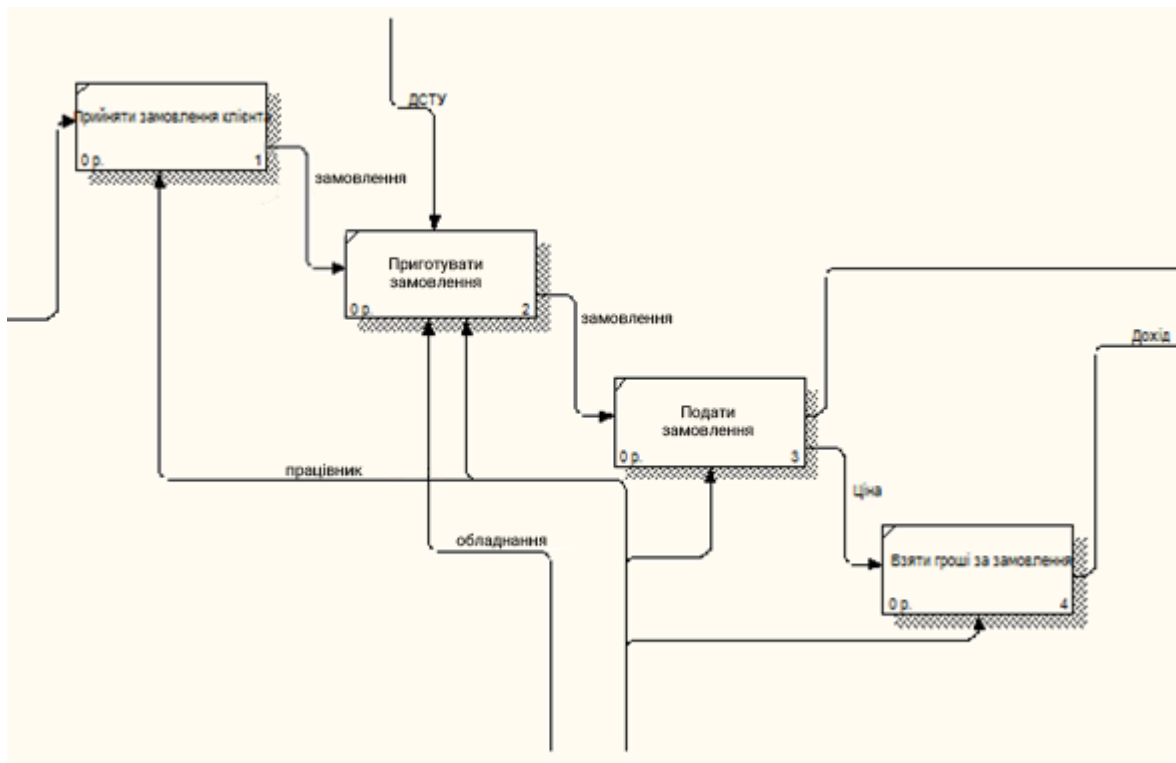


Рисунок 1.4 - Діаграма декомпозиції роботи «Приготування замовлення»

Також була створена SADT DFD діаграма(Рис 1.5) в якій показані тільки входи та виходи. DFD діаграми декомпозиції зображені на рисунках 1.6 - 1.7.

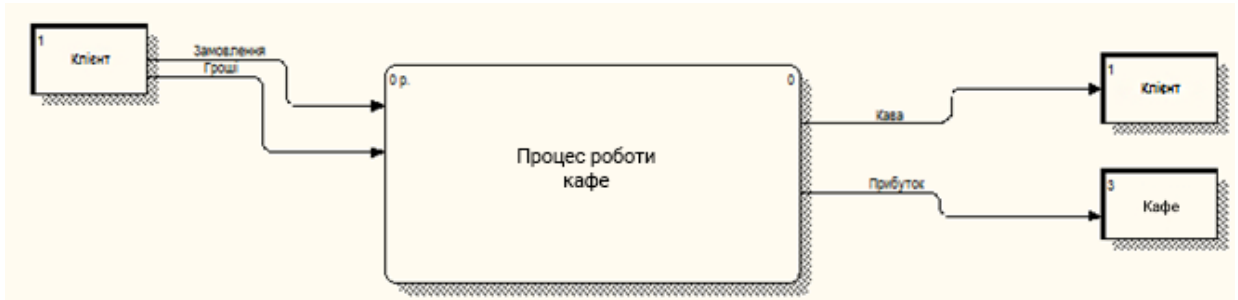


Рисунок 1.5 – DFD Контекстна діаграма кафе

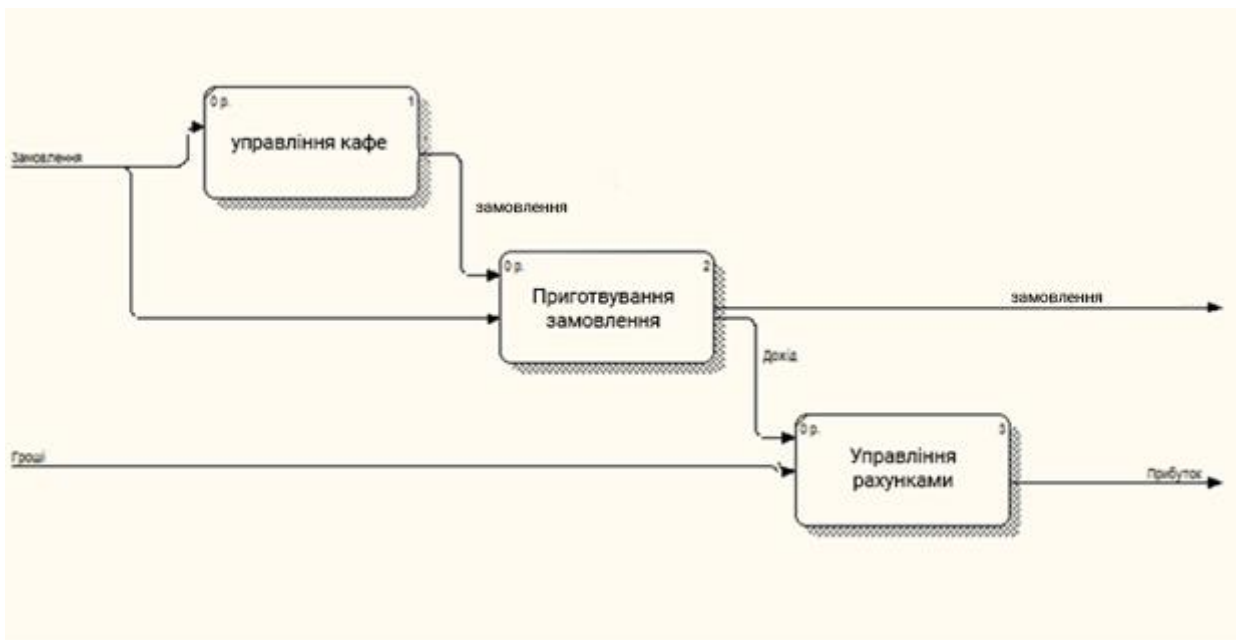


Рисунок 1.6 – DFD Діаграма декомпозиції контекстної діаграми кафе

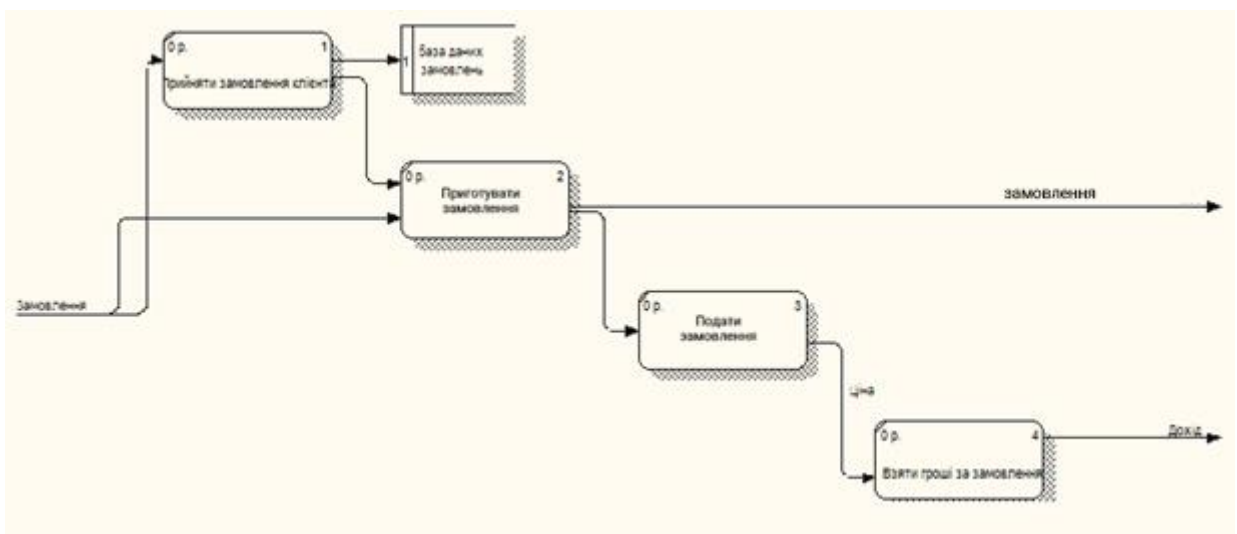


Рисунок 1.7 - Діаграма декомпозиції роботи «Приготування продукції»

Таким чином, функціональна модель має два рівні декомпозиції, розроблена з точки зору власника кафе. Це дало змогу виявити види робіт, які доцільно автоматизувати.

1.4. Аналіз існуючих бізнес-процесів

POS термінал - це електронний пристрій, що зчитує дані пластикової картки з магнітної смуги або чипу, розташованого на пластиковій картці, або зі смартфона з функцією NFC та встановленого відповідного ПЗ із внесеними реквізитами платіжної картки, і зв'язується з банком по електронних каналах зв'язку. Сума операції вводиться з клавіатури (якщо це POS-термінал, інтегрований до каси, то сума береться з даних каси до оплати). Всі дані операції друкуються на чеку терміналом. Наявність POS-терміналу дозволяє приймати до оплати всі типи міжнародних банківських карток, включаючи як найпоширеніші електронні картки «Visa» та «Mastercard», так і менш поширені у нашій країні China UnionPay та інші, що значно прискорює проведення операції оплати і скорочує термін відшкодування грошових коштів по проведених операціях.

1.4.1. HOREKER

POS система для кафе HORKERER розроблена з урахуванням потреб швидкого обслуговування гостей та ефективного управління бізнесом. Її особливості включають:

1. *Швидке обслуговування гостей:* Інтерфейс POS системи максимально простий та зручний для користувачів, що дозволяє персоналу швидко приймати замовлення та виконувати їх;
2. *Вбудований програмний PPO:* Система вже має вбудований програмний PPO, що спрощує фіскалізацію продажів та автоматичну передачу даних до податкової служби, забезпечуючи відповідність законодавству;

3. *Оперативне створення нових товарів:* Зручний інтерфейс дозволяє оперативно додавати нові товарні позиції, страви та напої без зайвих зусиль;
4. *Необмежена кількість товарних позицій:* POS система HORKERER не обмежує кількість товарів, які можуть бути додані до меню, що дозволяє адаптувати її до потреб кожної кафе;
5. *Легкий старт роботи:* Для початку роботи з системою не потрібно великої підготовки або інвестицій у дороге обладнання. Інтуїтивний інтерфейс дозволяє оперативно навчити персонал та почати обслуговування гостей;

Загалом, POS система HORKERER розроблена для максимальної ефективності та зручності управління кафе, дозволяючи зосередитися на обслуговуванні клієнтів та розвитку бізнесу.



Рисунок 1.8 – Програма HORKER

1.4.2. Sky Service

POS система для кафе Sky Service пропонує комплексне рішення для управління бізнесом, що включає ряд потужних інструментів для контролю, безпеки та прогнозування. Основні особливості цієї системи:

1. *Контроль усіх торгових точок у єдиному додатку*: Sky Service дозволяє керувати всіма кафеми в єдиному додатку. Ви можете створювати технологічні карти, категорії товарів і послуг, а також вибирати методи складського обліку: за останньою або середньою собівартістю для кожної окремої кафе;
2. *Безпека понад усе*: Система забезпечує високий рівень безпеки шляхом визначення прав доступу для кожного співробітника. Це гарантує, що кожен має доступ тільки до тих функцій, які необхідні для його роботи, забезпечуючи захист конфіденційної інформації;
3. *Прогноз на майбутнє*: Sky Service допомагає прогнозувати розвиток вашого бізнесу на тиждень, місяць або рік вперед. Завдяки цьому ви можете планувати необхідні ресурси, персонал та інші аспекти роботи кафе, щоб бути готовими до майбутніх викликів;
4. *Модуль складського обліку*: Програма включає модуль складського обліку, який формує рекомендації щодо закупівель. Це дозволяє вчасно замовляти всі необхідні продукти та товари, забезпечуючи безперебійну роботу кафе та задоволення клієнтів;

Загалом, POS система Sky Service – це інструмент, який надає все необхідне для ефективного управління кафе. Вона допоможе забезпечити високий рівень контролю, безпеки, прогнозування та управління складським обліком, що дозволяє зосередитися на розвитку бізнесу та обслуговуванні клієнтів.

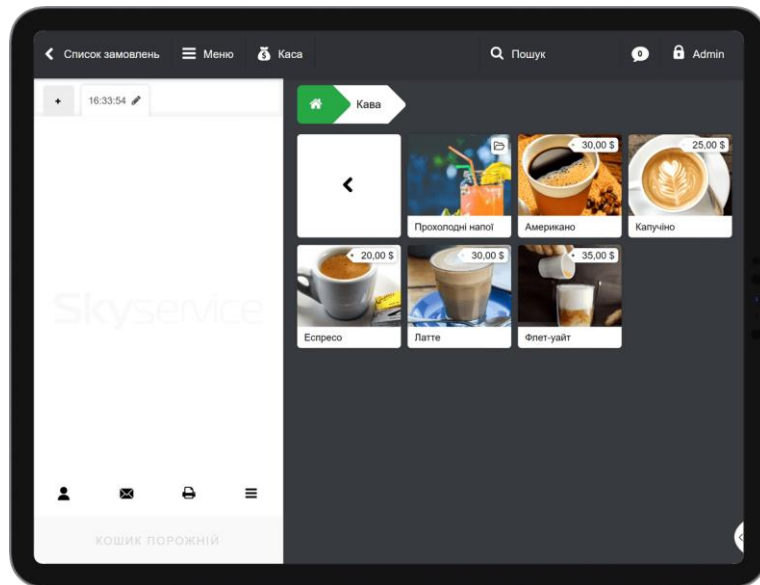


Рисунок 1.9 – Програма Sky Service

1.4.3. Poster

POS система для кафе Poster створена з урахуванням сучасних потреб бізнесу та забезпечує комплексний підхід до управління, звітності та інвентаризації. Основні особливості системи Poster включають:

1. *Інвентаризація, фіскалізація продажів та статистика:* Poster пропонує потужні інструменти для проведення інвентаризації, фіскалізації продажів та аналізу статистики. Це допомагає контролювати запаси, вести фінансову звітність та забезпечувати відповідність вимогам податкової служби;
2. *Фінансова звітність та маркетинг:* Система надає зручні інструменти для створення фінансових звітів та ведення маркетингових кампаній. Ви можете легко відслідковувати фінансові показники та запускати різноманітні акції для залучення клієнтів;
3. *Зручна каса на комп'ютері, планшеті та смартфоні:* Poster можна використовувати на різних пристроях, включаючи комп'ютери, планшети та смартфони. Це забезпечує гнучкість у роботі та дозволяє баристам приймати замовлення на ходу, скорочуючи час обслуговування клієнтів;

4. *Оформлення замовлень ще швидше:* Завдяки можливості приймати замовлення на планшеті біля стійки, бариста може одразу почати готувати напої, не затримуючи чергу в години пік. Це підвищує ефективність обслуговування та знижує час очікування для клієнтів;
5. *Приймайте оплату готівкою та карткою:* Poster підтримує різні методи оплати, включаючи банківські картки, готівку та часткову оплату бонусами. Це забезпечує зручність для клієнтів і полегшує процес прийому платежів;
6. *Працюйте й надалі без інтернету:* Навіть у разі втрати з'єднання з інтернетом, система обліку продовжує працювати, а дані синхронізуються, щойно з'явиться інтернет-з'єднання. Це дозволяє безперервно обслуговувати клієнтів навіть у разі перебоїв з інтернетом;
7. *Запускайте акції та накопичувальні системи:* Poster дає змогу запускати різні акції та бонусні програми, що допомагає збільшити обсяги продажів і залучити нових клієнтів, особливо в початковий період після відкриття кафе;

Загалом, POS система Poster надає широкий спектр інструментів для ефективного управління кафе, забезпечуючи гнучкість, зручність та надійність роботи.

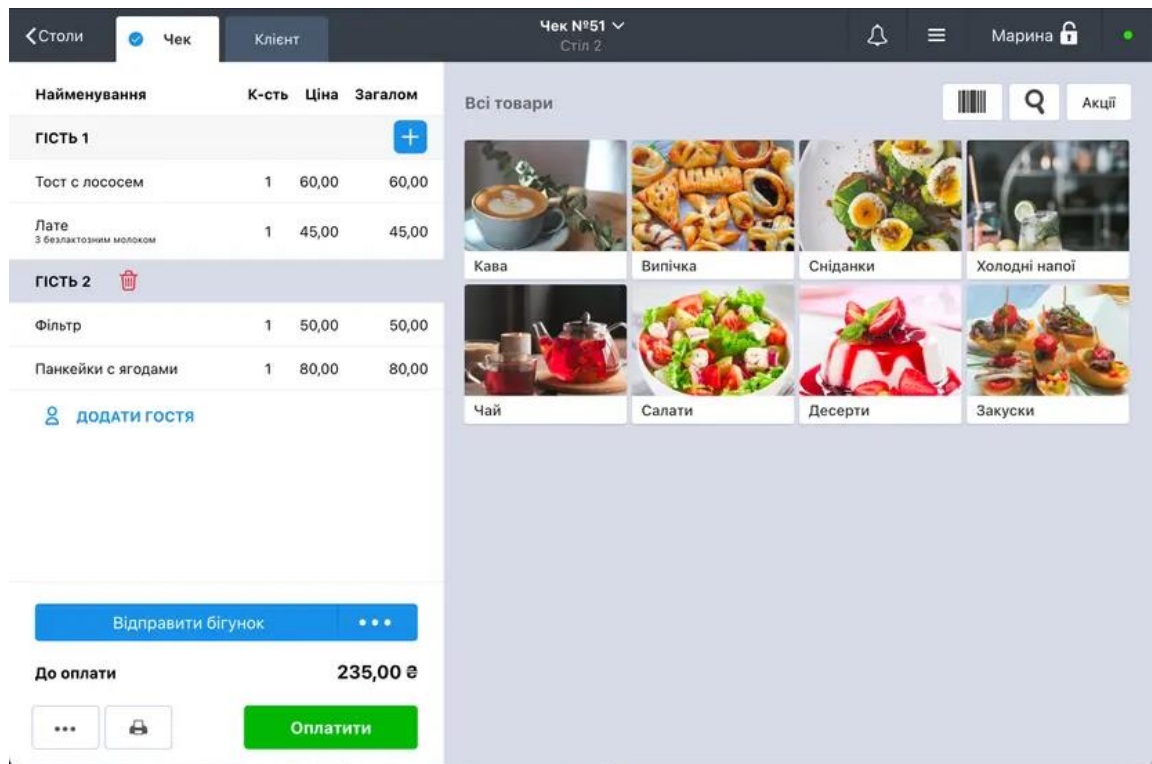


Рисунок 1.10 – Програма Poster

1.5. Порівняння існуючих бізнес-процесів

Мета розробки – створити систему інформаційної підтримки діяльності кафе під операційну систему Windows який надає можливість користувачам створювати, обробляти та видаляти замовлення.

Суть ідеї.

Сервіс представляє клієнтам можливість:

- створювати замовлення;
- редагувати та видаляти замовлення;
- розраховувати здачу та створювати чек;
- редагувати ціни на продукцію;
- слідити за об'ємом проданої продукції;
- автоматично створювати меню.

Опис продукту

Специфіка сервісу.

Система повинна надавати:

- перегляд замовлень;
- можливість створити чек замовлення;
- перегляд даних про продану продукцію;
- автоматичне створення меню кафе.

Опис технології та підтвердження потрібної кваліфікації.

Автоматизована система обробки замовлень кафе також повинна мати довідник продукції з можливістю змінити ціну на продукцію.

Система повинна мати високу швидкість роботи та відображати вірну інформацію.

Для реалізацію програмного продукту потрібно створити Windows програму. Для створення програми необхідні спеціалісти, що базуються на Windows Forms, SQL.

Основний вид діяльності за Класифікатором видів економічної діяльності - 62.01 Комп'ютерне програмування.

Наявність ліцензії (патентних прав).

Ліцензія MIT умовно-безкоштовного вільного використання[14].

Майбутній потенціал продукту.

Потенційна аудиторія – кафе які почали недавно працювати і яким потрібно вузькоспеціалізована автоматизована система що буде спрощувати обробку замовлень в кафе.

До потенційних проблем можна віднести те що є інші аналоги автоматизованих систем для кафе але на даний момент їх важко знайти та вони коштують дорожче. На таблиці 1.1 наведено Характеристику інновацій.

Таблиця 1.1. Характеристика інновацій

Ознаки класифікації	Види інновацій
1 За предметом інноваційної діяльності	Процесні
2 Масштаб новизни інновації	Не нові для підприємства
3 За місцем в системі (на підприємстві, у фірмі)	У фірмі
4 За сферою застосування	Технологічні
5 За глибиною внесених змін	
6 За задоволенням потреб	Задовольняють існуючу потребу
7 За напрямом реалізації	Призначені для продажу
8 За приналежністю до розроблення	Розроблені власними силами
9 За джерелами фінансування	Змішане фінансування

Місце реалізації проекту: Україна, власне обладнання.

Основні споживачі: Основними користувачами будуть кафе-початківці яким що хочуть зручно вести облік своїх замовлень. І мають персональний комп'ютер чи ноутбук на операційній системі Windows.

Конкурентне середовище:

- HORKER;
- Sky service;

В таблиці 1.2 показано Аналіз та порівняння програмних аналогів

Таблиця 1.2. Аналіз та порівняння програмних аналогів

Критерії	HORKER	Sky service	Розроблений додаток
Приємний і зрозумілий інтерфейс	+	+	+
Фільтрація за категоріями	+	+	+
Підтримка відео файлів	+	-	-
Підтримка файлів у форматі 2D	+	+	+
Створення чеку	+	+	+
Можливість автоматичного створення меню	-	-	+
Статистика продажу продукції	+	+	+
Адаптація під телефони та планшети	+	+	-

Перспективи ринку

Бізнес кафе один із самих поширених типів бізнесу, що створюється через відносно низький рівень стартового капіталу, тому сфера кафе поширена не тільки в Україні а ще й за кордоном.

План маркетингу.

Ідея даного додатку заснована на тому що при її придбанні клієнт отримує не тільки програму а ще й проект програми для внесення модифікацій.

При придбанні товару клієнт отримує права на модифікацію та продаж товару іншим бажаючим придбати вже модифіковану програму, при продажі модифікованого продукту виробнику основної програми йде відсоток від придбання. Таким чином клієнтами програми можуть стати не тільки малий бізнес а ще й розробники програмного продукту яким потрібна основа для модифікації та перепродажі додатку.

Стратегія просування товару на ринок, рекламна діяльність.

- розміщення оголошень на інтернет - ресурсах;
- установка банерів на відвідуваних і популярних сайтах;
- класичне просування у топ google за високо- та низькочастотними пошуковими запитами;
- просування через соціальні мережі.

Оцінка конкурентоспроможності програмного продукту

Моделювання конкурентоспроможності розробленого програмного продукту та його аналогу.

У процесі дослідження виділяються основні технічні та економічні параметри базового та нового варіантів програмного продукту. Також наводяться додаткові функції нового програмного продукту.

В таблиці 1.3 наведено характеристику основних техніко–економічних параметрів базового та нового варіантів.

Таблиця 1.3 - Характеристика основних техніко–економічних параметрів базового та нового варіантів програмного продукту

Назва параметру	Од. виміру	Варіант		Характеристика параметра нового варіанта відносно базового (↑, ↓ чи =)
		Базовий (аналог)	новий	
Вартість програмного продукту	грн	23880	15000	↑
Простота та зручність інтерфейсу	бали	6	9	↑
Кількість функцій програмного продукту	шт.	7	5	↓
Вага програмного продукту	мбайт	5,7(8)	1,9(10)	↑
Час виконання запиту	мс	5(10)	5(10)	=

Продовження таблиці 1.3

Можливість нарощування функціональних характеристик	бали	4	7	↑
Час відновлення системи після збою	сек	10(7)	2(10)	↑
Кількість людей, необхідних для обслуговування	чол.	1-2	1-2	=

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

- вартість програмного продукту;
- простота та зручність інтерфейсу;
- вага програмного продукту;
- можливість нарощування функціональних характеристик;
- час відновлення системи після збою.

За двома параметрами базовий та новий варіанти програмного продукту є ідентичними а саме:

- час виконання запиту;
- кількість людей, необхідних для обслуговування.

За параметром «кількість функцій програмного продукту новий варіант поступається базовому» .

Еталоном є точка багатовимірного простору (вектор), що утворена за таким правилом: серед показників-стимуляторів (здійснюється позитивний вплив на конкурентоспроможність) відбираємо дані з максимальним значенням, а серед показників де-стимуляторів (чинить негативний вплив на конкурентоспроможність) відбираємо дані з мінімальним значенням. Визначення еталонного значення наведено в таблиці 1.4.

Розрахунок інтегрального показника конкурентоспроможності базового і нового варіантів програмного продукту.

Інтегральний показник конкурентоспроможності відносно еталона ($I_{КПС}$) визначається за формулою 1.1:

$$I_{КПС} = \sum_{i=1}^n \frac{P_{\text{баз. (нов.)}i}}{P_{\text{еталон } i}}, \quad (1.1)$$

де $P_{\text{баз. (нов.)}i}$, $P_{\text{еталон } i}$ - величини за i -тим параметром відповідно базового, нового варіанта програмного продукту та еталона.

Проведені розрахунки будуть свідчити про те, що базовий та новий програмного продукту може переважати еталон за параметрами. Результатом буде одна із наступних умов: якщо $I_{КПС\text{баз}} > I_{КПС\text{нов}}$, то базовий програмного продукту перевищує новий зразок за конкурентоспроможністю, якщо $I_{КПС\text{баз}} < I_{КПС\text{нов}}$, то поступається йому, а при $I_{КПС\text{баз}} = I_{КПС\text{нов}}$, новий програмний продукт знаходиться на одному рівні з базовим.

Визначення ефективності нового ПП порівняно з базовим. Загальна ефективність виробництва нового ПП порівняно з базовим визначається за формулою 1.2:

$$E\Phi_{\text{заг.}} = E\Phi_{\text{min}} + E\Phi_{\text{max}}, \quad (1.2)$$

де $E\Phi_{\text{заг}}$ - загальна ефективність;

$E\Phi_{\text{min}}$ - мінімальна ефективність;

$E\Phi_{\text{max}}$ - максимальна ефективність.

$$E\Phi_{min} = \sum_{i=1}^n \left(1 - \frac{P_{нов.i}}{P_{баз.i}}\right), \quad (1.3)$$

де $E\Phi_{min}$ - мінімальна ефективність;

$P_{нов.i}$, $P_{баз.i}$ - величини за i -тим параметром відповідно нового та базового варіанту ПП. Проте, якщо серед параметрів є такі, для яких максимальне значення є найбільш ефективним, то рівняння за формулою 1.2 набуває наступного вигляду:

$$E\Phi_{max} = \sum_{i=1}^n \left(\frac{P_{нов.i}}{P_{баз.i}} - 1\right), \quad (1.4)$$

де $E\Phi_{max}$ - максимальна ефективність;

$P_{нов. i}$, $P_{баз i}$ - величини за i -тим параметром відповідно нового та базового варіанту програмного продукту.

Для розрахунку загальної ефективності визначимо відносне значення параметрів нового програмного продукту відносно базового. Розраховуємо ефективність окремого параметра нового програмного продукту за формулами 1.1 – 1.4, результати розрахунків заносимо до таблиці 1.4.

Таблиця 1.4. Визначення параметрів конкурентоспроможності програмного продукту та показника його ефективності

Назва параметра	Еталонне значення		Варіант		$\frac{P_{\text{нов } i}}{P_{\text{баз } i}}$	$1 - \frac{P_{\text{нов } i}}{P_{\text{баз } i}}$	$\frac{P_{\text{нов } i}}{P_{\text{баз } i}} - 1$
	max або min		$\frac{P_{\text{баз } i}}{P_{\text{еталон } i}}$	$\frac{P_{\text{нов } i}}{P_{\text{еталон } i}}$			
Вартість програмного продукту	Min	15000	-1,3	+1	0,6	0,4	
Простота та зручність інтерфейсу	Max	9	-1,14	+1	1,5	-0,5	
Кількість функцій програмного продукту	Max	7	-0,6	+1	0,7		-0,3
Вага програмного продукту	Min	1,9	+1	-1,12	0,3	0,7	
Час виконання запиту	Min	5	+1	+1	1		0

Продовження таблиці 1.4.

Можливість нарощування функціональних характеристик	max	7	-0,50	+1	1,75	-0,75	
Час відновлення системи після збою	Min	2	-1,5	+1	0,2	0,8	
Кількість людей, необхідних для обслуговування	min	1	-2	+1	1		0
УСЬОГО			-5,04	5,88		0,35	



Рисунок 1.11 – Номограма для порівняння параметрів конкурентоспроможності базового та нового варіантів програмного продукту

Висновок

В економічній частині бакалаврської роботи обґрунтовано доцільність розробки автоматизованої програмної системи.

При аналізі основних техніко-економічних характеристик було відмічено що новий програмний продукт за п'ятьма параметрами переважає аналог, а саме: вартість програмного продукту, простота та зручність інтерфейсу, можливість нарощування функціональних характеристик, час відновлення системи після збою. Однакові параметри були у часу виконання запиту та кількості людей необхідних для обслуговування. Новий продукт лише поступається в Кількості функцій програмного продукту.

Показник конкурентоспроможності нового програмного продукту (5,88) та його ефективність складає 102%

1.6. Розрахунок економічного ефекту від впровадження системи

В даному я врахую вартість усіх витрат, пов'язаних з проведенням автоматизації виробництва: вартість засобів автоматизації(табл 1.5) і монтажних матеріалів(табл 1.6), вартість електроенергії(табл 1.7) а також, економічний ефект.

Таблиця 1.5. Розрахунок вартості засобів автоматизації.

№ п/п	Назва обладнання	Один, вимір.	Кількість	Ціна, грн.	Вартість, грн.
1	ПК або ноутбук	Шт.	1	4000	4000
2	ОС	Шт.	1	4500	4500
3	POS-принтер	Шт.	1	2000	2000
4	Принтер	Шт.	1	3000	3000
	Всього				13500

Таблиця 1.6. Розрахунок вартості монтажних матеріалів.

№ п/п	Назва матеріалів	Один, вимір.	Кількість	Ціна, грн.	Вартість, грн.
1	Переноска електрична	Шт.	1	600	600
2	Кліпса	Шт.	100	100	100
3	Пластиковий хомут	Шт.	100	156	156
	Всього				856

Таблиця 1.7. Розрахунок вартості електроенергії

№ п/п	Назва	Один, вимір.	Кількість	Ціна, грн/год.	Вартість, грн/год.
1	Електроенергія силова	кВт/год.	0.3	2,64	0.792

Планований час роботи системи 8 год тобто за день система буде споживати 0,24 кВт/год тобто витрати становитимуть 6,336 грн на день.

Загальна сума витрат

$$13500 + 856 + 190,08 = 14546,08 \quad (1.1)$$

плюс 190,08 грн кожен місяць.

Встановлення системи автоматизації буде виконано техперсоналом кафе, встановлення системи не передбачає великих витрат тому витрати на встановлення не буде внесено.

Розрахунок економії від впровадження автоматизації.

В загальному випадку автоматизація виробництва може привести до наступних результатів:

- *росту продуктивності праці, зменшення часу обслуговування клієнтів, покращення обілку замовлень;*
- *покращення умов праці працівників;*

до спрощення технологічних схем, зменшення витрат на виробництво.

В даному випадку в після автоматизації, зменшення працівників немає, але пришвидшується процес обслуговування клієнтів та ведення обліку кафе. Отже, в даному випадку економії по фонду оплати праці не буде але буде пришвидшено обслуговування клієнтів.

Приблизний розрахунок обслуговування клієнтів до автоматизації, 480 хвилин - стандартний робочий день (8 год), 5 хвилин – приблизний час обслуговування вручну до автоматизації, $480/5=96$ клієнтів, Приблизний розрахунок обслуговування клієнтів після автоматизації, 480 хвилин - стандартний робочий день (8 год), 2 хвилини – приблизний час обслуговування після автоматизації, $480/2 = 240$ клієнтів, Коефіцієнт ефективності обслуговування

$$240/96 = 2.5 \quad (1.2)$$

Висновок

До впровадження системи обслуговували 96 клієнтів за день. Після впровадження системи обслуговують 240 клієнтів за день. Коефіцієнт ефективності становить 2.5, що означає збільшення продуктивності на 150%.

1.7. Висновок до розділу 1

Малі кафе часто стикаються з проблемами, пов'язаними з ефективним управлінням та обслуговуванням клієнтів. Впровадження автоматизованої системи дозволяє вирішити ці питання за рахунок автоматизації ключових бізнес-процесів, таких як облік замовлень, управління запасами, розрахунок вартості та обробка платежів. Це сприяє зниженню навантаження на персонал, скороченню часу обслуговування та підвищенню загальної продуктивності. Власна система дозволяє забезпечити більш високий рівень безпеки збереження інформації, що є особливо важливим у контексті захисту персональних даних клієнтів та фінансової інформації підприємства. Власне програмне забезпечення можна адаптувати до специфічних потреб бізнесу. Це дозволяє впроваджувати нові функції та

вдосконалювати існуючі, виходячи з вимог і змін на ринку. Можливість створення індивідуальних налаштувань, що враховують особливості кафе, допомагає забезпечити оптимальні умови для роботи та обслуговування клієнтів. Автоматизація кафе за допомогою розробки власної системи є важливим кроком для підвищення ефективності бізнесу та якості обслуговування клієнтів. Власне програмне забезпечення забезпечує більш високий рівень безпеки, гнучкості та можливості модифікації, що робить його вигідним інструментом для малих кафе, які прагнуть розвиватися та вдосконалювати свої процеси. Тому було прийнято рішення про розроблення автоматизованої системи для підтримки управління діяльністю кафе.

2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ

2.1. Загальні положення.

Найменування системи: «Інформаційна система для підтримки діяльності кафе»

Результати робіт зі створення системи оформлюються згідно з вимогами ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів у даному випадку визначається змістом і календарним планом виконання розробки.

У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватися і розвиватися.

2.2. Призначення і цілі створення системи.

Призначення системи.

Система призначена для автоматизації роботи кафе. Система автоматизує роботу кафе, оформлення замовлень та введення звітів. Також система містить дані про працівників та комплекс додаткових функцій, які дозволяють виконувати зручний введення різної інформації.

Цілі створення системи.

Основною метою створення системи є забезпечення оперативного оформлення замовлень, введення обліку замовлень, введення розкладу працівників. Це забезпечить створення умов для поліпшення діяльності кафе.

Задачі організаційного управління роботою кафе характеризуються високою складністю, комплексністю й не можуть повністю розв'язуватись ізольовано. Тому в умовах функціонування автоматизованої системи з'являється можливість виконання усіх завдань на сучасному рівні із забезпеченням точності, оперативності та достовірності інформації.

2.3. Характеристика об'єкта автоматизації.

Короткі відомості про об'єкт автоматизації.

Об'єктом автоматизації є діяльність типового кафе.

2.4. Вимоги до системи

Вимоги до системи в цілому.

Вимоги до структури і функціонування системи.

Система повинна мати клієнт-серверну архітектуру, що використовує єдину базу даних (надалі — БД).

Згідно з функціональною структурою кафе, система повинна бути пов'язана в мережі з автоматизованими робочими місцями:

- Власника;
- Працівника;
- Тех-персоналу;

Діагностування функціонування системи в мережі технікуму має передбачати виявлення відхилень від нормального процесу розв'язання задач і порушень у роботі комп'ютерно-технічних засобів, а також програмних помилок, забезпечуючи користувачів відповідними діагностичними повідомленнями.

Взаємозв'язок між підсистемами має здійснюватися на інформаційному рівні через загальну БД із використанням технічних засобів локальних комп'ютерних мереж.

Розвиток і модернізація системи повинні проводитися шляхом уточнення, нарощування чи заміни виконуваних функцій, модернізації технічних і програмних засобів по мірі розробки і впровадження нових поколінь комп'ютерів. Структура і технологія програмного забезпечення системи повинні забезпечити простоту їх модернізації та розвитку, з можливістю збільшення розмірності задач і масивів інформації, а також можливості реалізації їх на нових ПК.

Програмно-технічні засоби функціонування системи повинні мати програми з економіко-математичними та статистичними методами, методами моделювання, а також засоби табличного, текстового, графічного відображення даних. Програмна та інформаційна сумісність має забезпечуватися загальносистемним протоколом обміну, використанням проблемно-орієнтованих пакетів прикладних програм міжмашинних зв'язків і єдиною системою класифікації і кодування.

Функціонування системи має забезпечувати діалогову та мережну (розподілену) обробку даних.

Вимоги до чисельності і кваліфікації персоналу.

Персонал, який використовує автоматизовану систему, повинен дотримуватися наступних вимог:

- пройти навчання і отримати навички роботи на ПК;
- дотримуватись технологічних інструкцій при роботі з системою в діалоговому режимі;
- дотримуватись умов експлуатації ПК у відповідності з інструкціями по експлуатації;
- дотримуватись правил зберігання інформації і організації резервних копій БД;
- дотримуватись правил техніки безпеки при роботі на ПК.

Користувачами системи може виступати власник, працівник та тех-персонал. Вхід у систему повинен здійснюватися через пароль, який відображає рівень користувача: із правом коригування БД і без права. Залежно від рівня користувач отримує повний чи обмежений доступ до системи.

Показники призначення.

Показники призначення повинні характеризувати ступінь та якість автоматизації планової, інформаційно-облікової і управлінської діяльності кафе для його оптимального функціонування. Перелік і допустимі значення показників, при яких зберігається цільове призначення системи, повинні бути визначені на стадії техноробочого проектування.

Система повинна мати можливість налаштування на параметри об'єкта управління та периферійного обладнання при її модернізації та розвитку, а також зміні процесів та методів організаційного управління.

Вимоги до надійності.

Система є багатофункціональною і призначена для використання протягом робочого дня. Всі функції системи виконуються дискретно. У відповідності з ДСТУ 2226-93 оцінка надійності проводиться по кожній функції окремо. Враховуючи особливості функціонування системи, показники її надійності є показниками надійності СУБД, на якій вона реалізована, та технічних засобів, на яких вона експлуатується. Основними показниками надійності є:

L_i — ймовірність безвідмовного виконання задачі в заданий термін (імовірність того, що i -тий запит буде виконаний);

K_r — коефіцієнт готовності ПТК (програмно-технічного комплексу);

T_v — середній час відновлення ПТК;

T_e — мінімальний час між двома відмовами за календарний місяць.

Комплекс технічних засобів повинен передбачати:

- можливість запуску і розв'язання функціональних задач із різних робочих станцій;
- можливість переходу на локальний режим роботи.

Для забезпечення надійності програмного та інформаційного забезпечення необхідно передбачити використання:

- модульного, структурного і об'єктно-орієнтованого програмування;
- програмних засобів контролю вхідної інформації з видачею користувачу повідомлень про виявлені помилки;
- програмних засобів коригування для виявлення і виправлення помилок у БД;
- засобів захисту від збоїв, несанкціонованого доступу, помилкових дій персоналу і т.д.;
- резервних копій БД.

Вимоги до безпеки.

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи потрібно дотримуватись вимог ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; по доступним рівням освітленості, вібраційних і шумових навантажень слід дотримуватися вимог відповідно ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

Вимоги з ергономіки та технічної естетики.

Загальні ергономічні і естетичні вимоги до системи повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Освітленість робочого місця повинна відповідати ДСТУ EN 12464-1:2016, ДБН В.2.5-28-2006.

Засоби відображення повинні розміщуватися таким чином, щоб кут спостереження екрану складав не більше, ніж 45 градусів, мінімальна відстань спостереження екрану — 0,3 м, рекомендована — 0,5 м.

При розробленні ПЗ слід створити зручний інтерфейс для запобігання втомлюваності користувача.

Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи.

Види обслуговування системи визначаються у відповідності з ДСТУ EN 13306:2019. Загальні вимоги по експлуатації, технічному обслуговуванню і ремонту повинні відповідати ДСТУ 3576-97.

Для розміщення технічних засобів системи необхідні площі, визначені в ДБН В.2.2-9-2009. При цьому слід дотримуватися вимог, зазначених в експлуатаційній документації. Напруга живлення технічних засобів системи 220/380 В змінного струму, частотою (50 ± 1) Гц. Допустиме відхилення напруги від +10 до -15%, тривалість перерв у живленні не повинна перевищувати 0,001 с.

Кількість, кваліфікація і режими роботи обслуговуючого персоналу повинні відповідати рекомендаціям, зазначеним в технічних умовах і інструкціях з експлуатації окремих ТЗ.

Склад, розміщення і умови зберігання компонентів технічних засобів системи визначається рекомендаціями, зазначеними в експлуатаційній документації на ці елементи.

Регламент обслуговування повинен відповідати їх рівню і умовам роботи, щоб у випадку відмови системи забезпечити роботу в аварійному режимі.

Вимоги до захисту інформації від несанкціонованого доступу.

Для надійності збереження і доступу до інформації необхідно використовувати засоби захисту:

- 1) серверних операційних систем Windows;
- 2) локальної мережі та програми захисту в мережі Firewall.
- 3) клієнт-серверної СУБД:
 - тригери, представлення;
 - процедури та функції;
 - встановлення груп користувачів і ролей використання.

Крім цього, кожен сеанс роботи системи має розпочинатися з введення індивідуального паролю. Система парольного захисту повинна мати власні засоби періодичної зміни паролів або використовувати стандартні засоби середовища розроблення. Для надійного захисту від несанкціонованого доступу кожен із працівників повинен мати персональний пароль. Крім того, деякі таблиці треба захистити від можливого редагування, доповнення чи вилучення інформації.

Вимоги щодо збереження інформації при аваріях.

Необхідно передбачити засоби резервного збереження БД в архіві після коригування і можливість завантажити БД з архіву у випадку її руйнування.

Резервний архів і БД мають знаходитись на різних машинних носіях чи пристроях.

Вимоги по захисту від впливу зовнішніх діянь.

Електрична складова електромагнітного поля завад в приміщеннях не повинна перевищувати $0,3 \text{ В/м}^2$ в діапазоні частот від 0,15 до 300 МГц. Для

захисту від впливу електромагнітних полів та індустриальних завад слід передбачити різноманітні екрани та фільтри.

Засоби, які виключають вплив шкідливих факторів на функціонування комплексу технічних засобів, повинні бути запроектовані згідно з ДБН В.2.2-9-2009. Обчислювальні засоби по стійкості до зовнішніх впливів повинні відповідати ДСТУ 2506-94.

Вимоги до патентної чистоти.

При створенні даної системи патентні дослідження не проводяться.

Вимоги по стандартизації і уніфікації.

У системі кодування інформації необхідно проводити за світовим класифікатором і стандартом.

Вимоги до функцій.

Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 2.1.

Функції мають забезпечити раціональну організацію роботи користувача на основі безперервної технології: заповнення БД, довідників, формування різноманітних звітів і виконання інших функцій, визначених чинним документом. При цьому пріоритетом є зручність введення та використання інформації користувачем за рахунок формування підказок і меню на екрані монітора.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

№ п/ п	Найменування функції	Вхідна інформація	Вихідна інформація
1	Створення, редагування та оформлення замовлення	Таблиці БД «Order», «Order Details», «Production»	Форма каса з замовленнями
2	Формування та виведення каталогу Постачальників	Таблиці БД «Supplier»	Форма зі списком Постачальників
3	Формування та виведення каталогу Працівників	Таблиці БД «Employees»	Форма зі списком Працівників
4	Формування та виведення каталогу Посад	Таблиці БД «Positions»	Форма зі списком Посад
5	Формування та виведення каталогу Продукції	Таблиці БД «Production»	Форма зі списком Продукції
6	Формування та виведення Розкладу	Таблиці БД «Schedule»	Форма зі Розкладом роботи працівників
7	Формування та виведення Типу продукції	Таблиці БД «Type Of production»	Форма зі списком типів продукції
8	Виведення Статистики	Таблиці БД «Order», «Order Details», «Production»	Чарти «Кількість проданої продукції» «Сума продажу за місяць» «Створено замовлень»
9	Створення Звітів та Чеку	Таблиці «Order», «Order Details», «Production», «Schedule»	Створення Звітів Всіх замовлень або одного замовлення, а також чеку замовлення

У вимогах до математичного забезпечення (МЗ) система не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраної СУБД.

Вимоги до інформаційного забезпечення (ІЗ).

Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. ІЗ повинно гарантувати раціональну організацію зберігання інформації та доступу до неї.

Заповнення БД інформацією покладається на замовника за методиками і формами, створеними розробниками системи.

Склад, структура і спосіб організації інформації представляються у логічній моделі БД і можуть уточнюватись на етапі технічного проектування.

Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні системи — використання резервних копій БД.

Вимоги до лінгвістичного забезпечення (ЛЗ).

Для розробки програмних засобів, які реалізують виконання функцій і забезпечують сервіс користувачів повинні використовуватися мови високого рівня, які забезпечують створення структурних програм, а також мова обраної СУБД для здійснення доступу та маніпулювання даними.

Організація діалогу користувача до системи має будуватися на наборах меню і підказок, орієнтованих на виконання користувачем функцій. Запити користувача до системи повинні задаватись переважно природною мовою.

Вимоги до програмного забезпечення (ПЗ).

Загальносистемне ПЗ має забезпечувати надійне і якісне виконання функціональних завдань системи. До загальносистемного ПЗ належить:

- операційна система (далі ОС) — Windows
- система управління БД (далі СУБД) — MySQL .

Загальні вимоги до системного ПЗ можна сформулювати так:

- мінімальні вимоги до ресурсів технічних засобів (ТЗ);
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань системи.

Вимоги до ОС:

- мінімальне використання ресурсів комп'ютера для власних потреб, передусім оперативної і дискової пам'яті;
- максимальна швидкодія при управлінні зовнішніми пристроями;
- ОС сервера — Windows, ОС клієнта — Windows

Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- мінімальні вимоги до ТЗ.

Програмні засоби введення та виведення даних і ведення діалогу повинні забезпечувати:

- виведення необхідних даних на екран у вигляді відповідних відеограм;
- супровід введення даних контролем і сигналізацією користувачу про наявність помилок з можливістю їх виправлення під час введення даних;
- керований комп'ютером діалог при введенні даних;
- виведення даних у відповідному вигляді (формі документа) за запитом користувача.

При розробленні спеціального ПЗ слід виконати наступні вимоги:

- використані програми мають бути сумісні між собою та із загальносистемним ПЗ;
- ПЗ має розроблятися засобами об'єктно-орієнтованого програмування;
- забезпечити відповідність інтерфейсу користувача стандартам Windows;
- необхідна модульна структура програм;

- повинна бути передбачена можливість розширення складу задач у відповідності з новими функціональними потребами;
- ПЗ не повинно залежати від типу зовнішніх пристроїв (принтерів, дисків, сканерів тощо);
- діалог із користувачем повинен проводитись за допомогою клавіатури або миші з поясненням виконання дій і можливістю отримання підказки.

Вимоги до технічного забезпечення.

Технічні засоби системи (табл. 2.2.) повинні забезпечувати виконання функцій, перерахованих в таблиці 2.1(табл 2.1.).

Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах.

Таблиця 2.2. Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики комп'ютера
Технічне забезпечення для сервера	
1	HP ML115 Intel Xeon Quad Core 2,5 GHz\8 Gb\1 TB RAID5\ LAN 1 Gbit
Технічне забезпечення для клієнта	
1	Athlon QL-65 Dual Core 2,1 GHz; RAM: 2048 Mb; HDD: 250 Gb;
2	Монітор 15"
3	Миша USB
4	Клавіатура USB

Вимоги до метрологічного забезпечення.

Система не має вимірювальних каналів, вимірювального обладнання і приладів, тому вимоги до даного виду забезпечення не висуваються.

Вимоги до організаційного забезпечення.

Організаційне забезпечення системи розробляється в відповідності з вимогами державного стандарту по АСУП.

При впровадженні системи не передбачається збільшення штатної чисельності підприємства. Територіальне розміщення робочих місць, на яких буде встановлена система, визначається підприємством.

До функціонування системи висуваються наступні вимоги:

- наказом директора визначається список співробітників, які мають доступ до системи;
- контроль і прийняття рішень при аварійних ситуаціях при експлуатації системи здійснює відповідальний за систему.

2.5. Склад і зміст робіт по створенню системи

Стадії створення системи і терміни виконання робіт наведені в таблиці 2.3(табл 2.3.).

Таблиця 2.3. Найменування робіт при створенні системи

№ п/п	Найменування робіт	Строки виконання робіт
1	Передпроектне дослідження об'єкта автоматизації	20.04.2024
2	Технічне завдання	30.04.2020
3	Технічний проект	20.05.2020
4	Оформлення документації	23.05.2020

2.6. Порядок контролю і приймання системи

Система вводиться на діючому ДП кафе. При введенні в дію система повинна пройти приймальні випробування згідно з ДСТУ 3974-2000.

Випробування для визначення працездатності і рішення про можливість приймання системи в дослідну експлуатацію проводять розробники разом із замовником. Програму випробувань складає розробник і затверджує замовник.

Здача в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача. За результатами дослідної експлуатації формується перелік доробок і рекомендовані строки їх виконання.

Введення в дію системи оформлюється актом здачі-прийому.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- проводить укомплектування технічних засобів;
- організовує навчання користувачів системи роботи на ПК і вивчення інструкції з її експлуатації;
- проводить досліду експлуатацію і вводить систему в дію.

2.8. Вимоги до документації

На систему розробляється комплекс документації у складі: технічне завдання та технічний проект.

Документація на систему розробляється у відповідності з вимогами Державних стандартів серії 19 «Єдина система програмної документації» та серії 24 «Єдина система стандартів автоматизованих систем управління».

2.9. Джерела розробки

При розробленні технічного завдання на систему використано наступні документи:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Інформаційне забезпечення системи

Для створення моделі був використаний CASE-засіб СА ERwin Process Modeler. На основі виявлених проблем була розроблена логічна модель (Додаток А, Рис. А1.) за допомогою CASE-засобу СА ERwin Data Modeler. Логічна модель дозволяє відобразити об'єктну декомпозицію підприємства, незалежно від використовуваної СУБД.

На основі створеної логічної моделі була розроблена фізична модель (Додаток А, Рис. А2.), і лише після цього генерується системний код. Між функціональною моделлю та моделлю даних встановлюється зв'язок, який визначає відносини та набір даних у сутностях. В ERwin на логічному рівні модель відображається у вигляді так званої сутності (що відповідає звичайній таблиці у фізичній моделі), яка містить атрибути (колонки таблиці).

Розроблена модель кафе містить у собі набір із 8 таблиць:

«Supplier»(Таблиця 3.1.), «Production»(Таблиця 3.2.), «Order_Details»(Таблиця 3.8.), «Type_of_production»(Таблиця 3.3.), «Order»(Таблиця 3.7.), «Employees»(Таблиця 3.4.), «Positions»(Таблиця 3.5.), «Schedule»(Таблиця 3.6.)

Таблиця 3.1. Таблиця Supplier

Назва	Тип	Значення
SupplierID	int	Ід постачальника
Supplier_Name	varchar(50)	Назва постачальника
FIO	varchar(50)	ПІБ постачальника
Phone	varchar(50)	Номер телефону постачальника
Email	varchar(50)	Електронна пошта постачальника

Таблиця 3.2. Таблиця *Production*

Назва	Тип	Значення
ProductID	int	Ід продукту
SupplierID	int	Ід постачальника
ProductName	varchar(50)	Назва продукту
Price	int	Ціна
Type_of_production	int	Тип продукту

Таблиця 3.3. Таблиця *Type_of_production*

Назва	Тип	Значення
Type_ID	int	Ід типу продукту
Product_name	varchar(50)	Назва типу продукту

Таблиця 3.4. Таблиця *Employees*

Назва	Тип	Значення
Employee_ID	int	Ід працівника
FIO	varchar(50)	ПІБ працівника
ID_position	int	Ід посади
username	varchar(50)	Логін
password	varchar(50)	Пароль

Таблиця 3.5. Таблиця *Positions*

Назва	Тип	Значення
ID_position	int	Ід посади
Position	varchar(50)	Посада

Таблиця 3.6. Таблиця Schedule

Назва	Тип	Значення
Employee_ID	int	Ід працівника
FIO	varchar(50)	ПІБ працівника
Monday	varchar(50)	Час роботи працівника
Tuesday	varchar(50)	Час роботи працівника
Wednesday	varchar(50)	Час роботи працівника
Thursday	varchar(50)	Час роботи працівника
Friday	varchar(50)	Час роботи працівника
Saturday	varchar(50)	Час роботи працівника
Sunday	varchar(50)	Час роботи працівника

Таблиця 3.7. Таблиця Order

Назва	Тип	Значення
OrderID	int	Ід замовлення
Date_of_order	date	Дата замовлення
Status_of_order	bit	Статус замовлення
Employee_ID	int	Ід працівника

Таблиця 3.8. Таблиця Order_Details

Назва	Тип	Значення
SubOrderID	int	Ід позиції в замовленні
OrderID	int	Ід замовлення
ProductID	int	Ід продукту
Amount	int	Кількість продукту

Логічна та фізична модель, створена в Erwin Data Modeler була автоматично згенерована як база даних в Microsoft SQL server, включаючи не тільки таблиці з атрибутами, але й первинні та зовнішні ключі, зв'язки між таблицями. Запуск MS SQL Serverта аутентифікація користувача(рис 3.1), створення нової бази даних в середовищі MS SQL(рис 3.2), імпорт фізичної моделі з Erwin в SQL Server(рис 3.3) та вид бази даних(рис 3.4) показано на рисунках нижче.

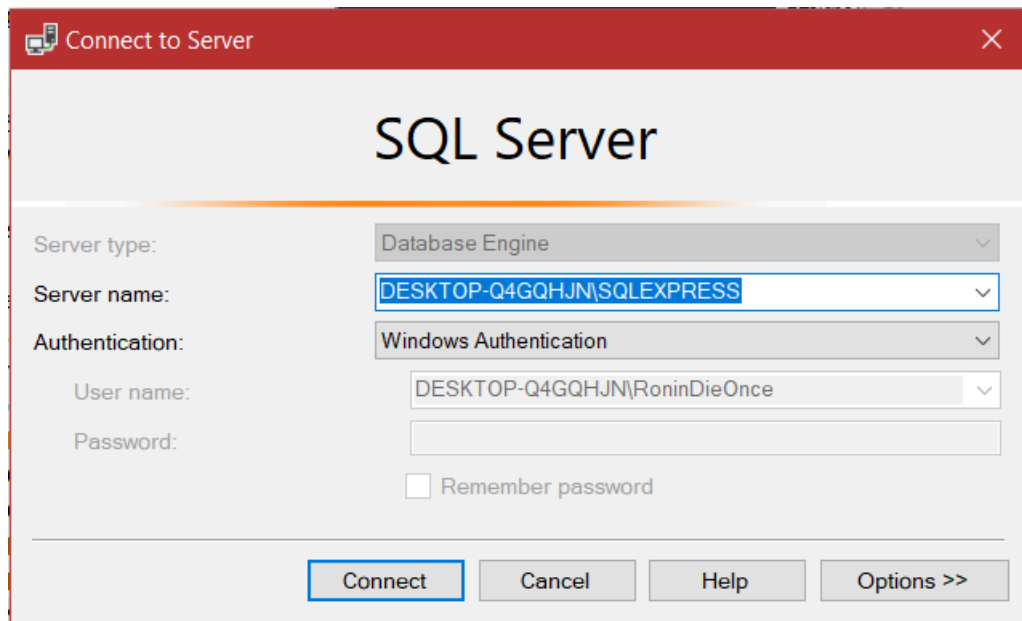


Рисунок 3.1 – Авторизація користувача SQL Server

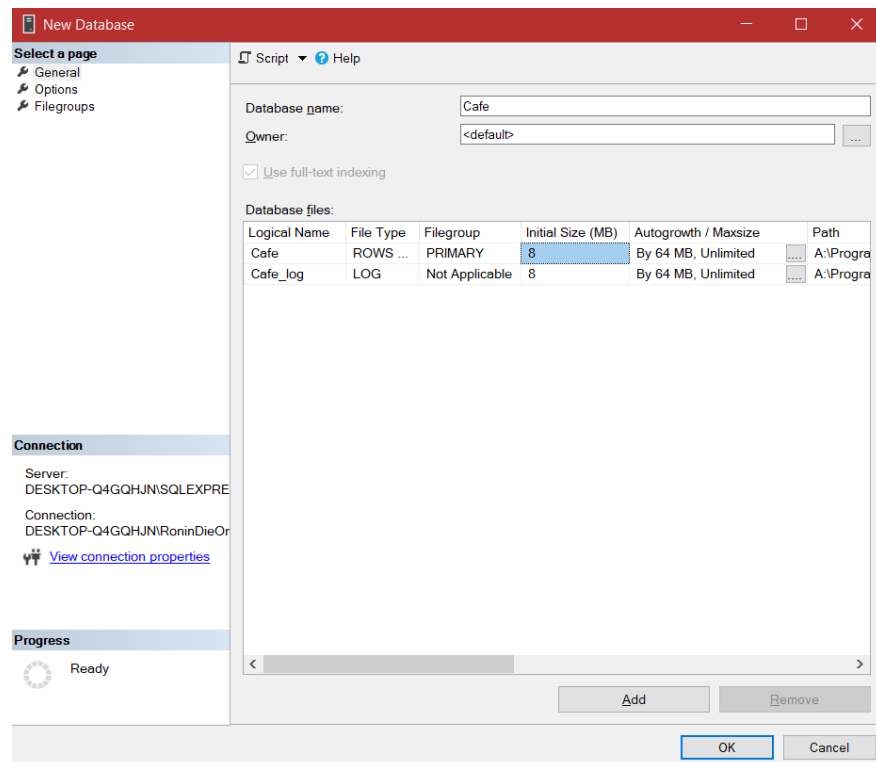


Рисунок 3.2 – Створення нової БД

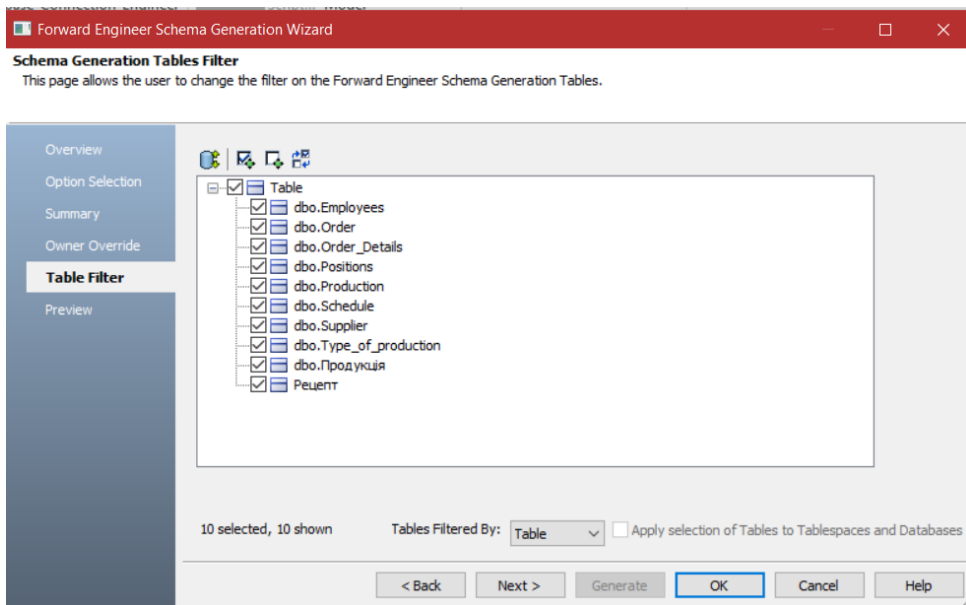


Рисунок 3.3 – Імпорт фізичної моделі в SQL Server

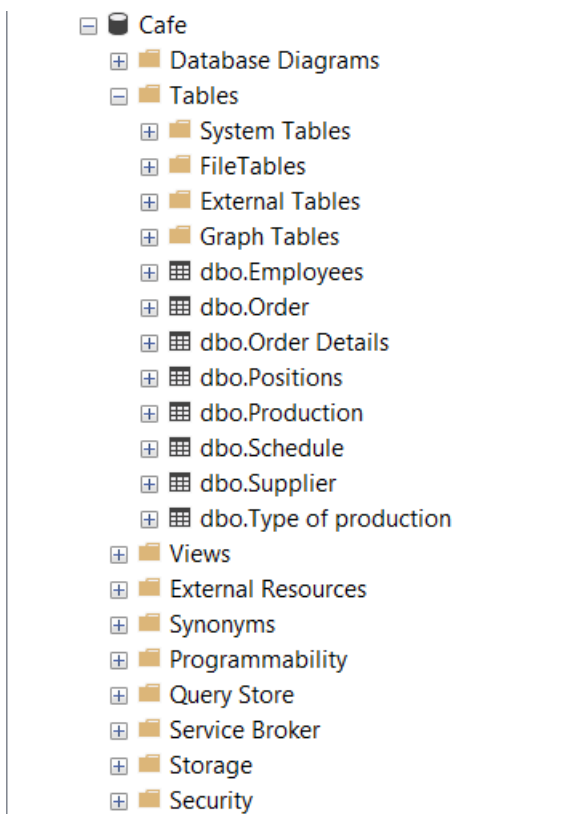


Рисунок 3.4 – Імпортована база даних

Діаграма імпортованої бази даних наведено в додатку (Додаток А, Рис. А3.),

Всі таблиці зображено на рисунку 3.5.

Table Name	Column Name	Data Type	Allow Nulls
DESKTOP-Q4GQHJN\...-e - dbo.Supplier	SupplierID	int	<input type="checkbox"/>
	Supplier_Name	varchar(50)	<input checked="" type="checkbox"/>
	FIO	varchar(70)	<input checked="" type="checkbox"/>
	Phone_Number	varchar(15)	<input checked="" type="checkbox"/>
	Email	varchar(50)	<input type="checkbox"/>
DESKTOP-Q4GQHJN\...-e - dbo.Production	ProductID	int	<input type="checkbox"/>
	SupplierID	int	<input checked="" type="checkbox"/>
	ProductName	varchar(50)	<input checked="" type="checkbox"/>
	Price	int	<input checked="" type="checkbox"/>
	Type_of_production	int	<input type="checkbox"/>
DESKTOP-Q4GQHJN\...-e - dbo.Type of production	Type_ID	int	<input type="checkbox"/>
	Product_name	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
DESKTOP-Q4GQHJN\...-e - dbo.Positions	ID_position	int	<input type="checkbox"/>
	Position	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
DESKTOP-Q4GQHJN\...-e - dbo.Schedule	Employee_ID	int	<input type="checkbox"/>
	FIO	varchar(50)	<input checked="" type="checkbox"/>
	Monday	varchar(50)	<input checked="" type="checkbox"/>
	Tuesday	varchar(50)	<input checked="" type="checkbox"/>
	Wednesday	varchar(50)	<input checked="" type="checkbox"/>
	Thursday	varchar(50)	<input checked="" type="checkbox"/>
	Friday	varchar(50)	<input checked="" type="checkbox"/>
	Saturday	varchar(50)	<input checked="" type="checkbox"/>
Sunday	varchar(50)	<input checked="" type="checkbox"/>	
DESKTOP-Q4GQHJN\...-e - dbo.Order	OrderID	int	<input type="checkbox"/>
	Date_of_order	date	<input checked="" type="checkbox"/>
	Status_of_order	bit	<input checked="" type="checkbox"/>
	Employee_ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
DESKTOP-Q4GQHJN\...-e - dbo.Order Details	SubOrderID	int	<input type="checkbox"/>
	OrderID	int	<input checked="" type="checkbox"/>
	ProductID	int	<input checked="" type="checkbox"/>
	Amount	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 3.5 – загальний вид таблиць

3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Перед створенням інтерфейсу було створено орієнтований дизайн форми каси(Додаток С, Рис. С.1.).

Для створення нового проекту в середовищі Visual Studio 2022 створюємо новий проект Windows Forms App (.Net Framework) та заходимо в Data explorer щоб з'єднати бд з середовищем (рис 3.6).

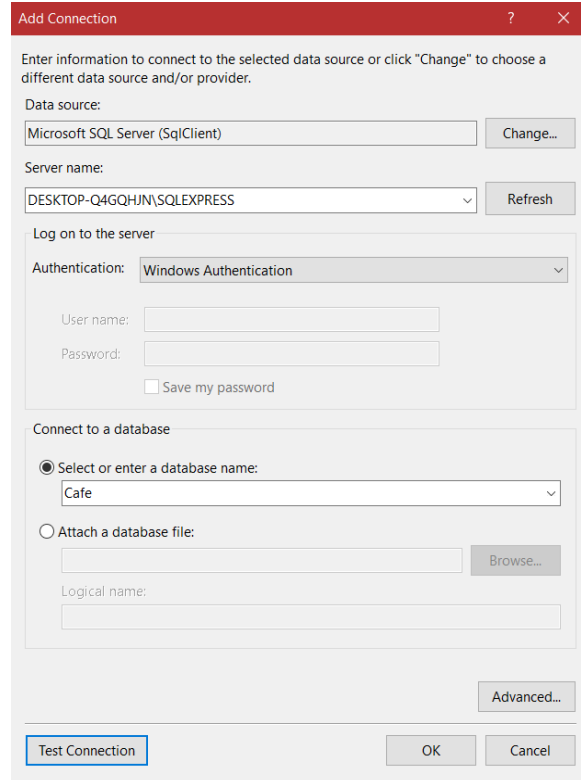


Рисунок 3.6 – з'єднання бд з середовищем VS 2022

Та перевіряємо з'єднання натиснувши кнопку «Test Connection»(рис 3.7)

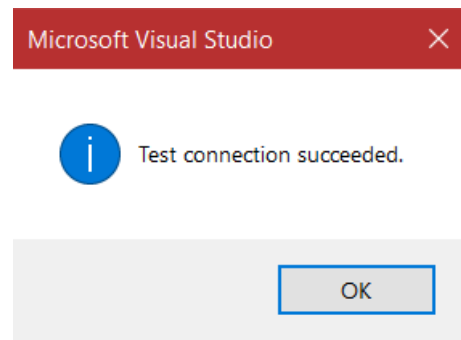


Рисунок 3.7 – Успішна перевірка з'єднання

Додаємо Data Source в відповідному вікні(рис 3.8).

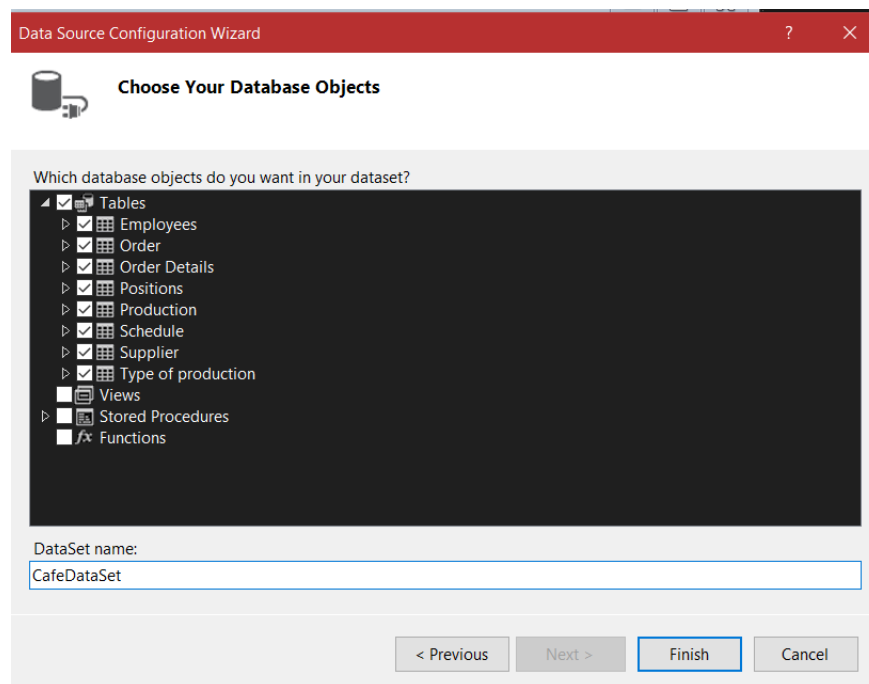


Рисунок 3.8 – Додавання нового Data Source

Для створення Інтерфейсу використовувались наступні елементи:

- *Button*: Кнопка це інтерактивний елемент інтерфейсу, який виконує певну дію при натисканні. Вона зазвичай має прямокутну форму з текстом, іконкою або їх комбінацією. Текст на кнопці короткий і описовий, наприклад, "Зберегти" або "Надіслати", що вказує на дію. Кнопки можуть мати різний колір, розмір і стиль залежно від контексту використання. Вони змінюють свій вигляд при натисканні, сигналізуючи про успішну взаємодію. Колір часто вказує на важливість або тип дії: червоний для видалення, зелений для підтвердження. Кнопки можуть бути неактивними, якщо дія недоступна, або змінювати стан залежно від контексту;

- *Textbox*: Текстове поле це інтерактивний елемент інтерфейсу, призначений для введення та редагування текстової інформації користувачем. Воно зазвичай має прямокутну форму з межами, які виділяють його на екрані. Текстове поле може містити підказку (placeholder), що зникає при введенні тексту. При фокусуванні на текстовому полі, межі можуть змінювати колір або стиль, сигналізуючи про готовність до введення даних. Текстові поля використовуються у формах, пошукових рядках та інших місцях, де потрібно ввести текстову інформацію.
- *Label*: Мітка це статичний елемент інтерфейсу, який використовується для відображення текстової інформації або пояснень для інших елементів інтерфейсу, таких як текстові поля, кнопки або перемикачі. Мітка зазвичай розташована поруч з цими елементами і надає додаткову інформацію або інструкції для користувача. Вона може містити текст різного стилю та розміру, а також може бути пов'язана з іншими елементами, щоб покращити доступність та зручність використання інтерфейсу;
- *Panel*: Панель це контейнерний елемент інтерфейсу, який використовується для групування та організації інших елементів інтерфейсу в межах одного візуального блоку. Вона може містити різні типи елементів, такі як кнопки, текстові поля, мітки, списки та інші панелі, забезпечуючи логічне та структуроване розташування контенту. Панель часто використовується для поділу інтерфейсу на окремі секції, полегшуючи навігацію та взаємодію користувача з додатком або веб-сайтом. Панелі можуть мати різні стилі, кольори та розміри, відповідно до дизайну та функціональних вимог інтерфейсу;
- *PictureBox*: Поле зображення це елемент інтерфейсу, який використовується для відображення зображень у додатку або на веб-сторінці. Воно підтримує різні формати зображень, такі як JPEG, PNG, GIF та інші. Поле зображення зазвичай має прямокутну форму, але може

бути налаштоване відповідно до вимог дизайну. Цей елемент дозволяє відобразити статичні зображення, а також може підтримувати динамічне завантаження зображень з різних джерел, таких як файли на сервері або інтернет-адреси. Поле зображення часто використовується в галереях, профілях користувачів, рекламних банерах та інших місцях, де потрібно показати візуальний контент. Крім відображення, воно може підтримувати масштабування, обрізку та інші способи обробки зображень для покращення користувацького досвіду;

— *MenuStrip*: Меню - це елемент інтерфейсу, який використовується для організації та надання користувачам доступу до різних команд і функцій додатку через структуровані списки. Меню зазвичай розташовується у верхній частині вікна програми або веб-сторінки, але може бути й у інших місцях інтерфейсу. Меню складається з декількох пунктів, які можуть містити підменю, утворюючи ієрархічну структуру. Пункти меню можуть містити текст, іконки або комбінацію обох для кращої візуалізації доступних дій. При наведенні курсору або натисканні на пункт меню, відкривається список доступних команд або опцій. Цей елемент інтерфейсу часто використовується для зручного групування пов'язаних функцій, таких як "Файл", "Правка", "Вид", "Довідка" тощо, що дозволяє користувачам легко знаходити та виконувати необхідні дії. Меню підтримує різні методи взаємодії, включаючи клавіатурні скорочення, що підвищує ефективність роботи користувача;

— *Chart*: Діаграма це графічний елемент інтерфейсу, який використовується для візуалізації даних у вигляді графіку, графіку або іншого типу візуального представлення. Вона дозволяє користувачам легко аналізувати та розуміти великі обсяги даних, порівнювати значення, виявляти тенденції та залежності. Діаграми можуть бути різних типів, таких як лінійні, стовпчасті, кругові, точкові тощо, в залежності від характеру даних та мети візуалізації. Кожен тип діаграми має свої власні особливості та способи відображення інформації.

- Діаграми зазвичай можуть бути налаштовані з використанням різноманітних параметрів, таких як колір, розмір, шрифт тощо, щоб підлаштуватися під вимоги конкретного проекту або дизайну інтерфейсу. Цей елемент інтерфейсу часто використовується для відображення аналітичної інформації, звітів, статистики, тенденцій розвитку та інших даних, що дозволяє користувачам швидко зрозуміти сутність представленої інформації та приймати обґрунтовані рішення;
- *ListBox*: Список це елемент інтерфейсу, призначений для відображення переліку елементів, з якими користувач може взаємодіяти. Кожен елемент списку зазвичай містить текстову інформацію або інші візуальні елементи, які представляють деякі об'єкти або дані. Список може бути відображений у вигляді вертикального або горизонтального списку з прокруткою, якщо кількість елементів перевищує видиму область. Користувач може прокручувати список, щоб переглянути всі доступні елементи, а також вибирати один або декілька елементів для подальшої обробки. Список може бути статичним, коли вміст задається наперед в коді програми, або динамічним, коли вміст генерується динамічно на основі даних з бази даних, файлу або іншого джерела. Цей елемент інтерфейсу часто використовується для відображення списків варіантів вибору, пунктів меню, навігаційних елементів, списків файлів та багатьох інших сценаріїв взаємодії з користувачем;
 - *DateTimePicker*: Вибір дати та часу це інтерактивний елемент інтерфейсу, призначений для введення та вибору дати та/або часу користувачем. Елемент зазвичай має вигляд текстового поля або кнопки, яку можна клацнути, щоб відкрити календар або вибрати часовий пікер для вибору дати або часу. Вибір дати та часу може бути здійснений шляхом вибору конкретної дати на календарі або введення дати вручну в текстове поле. Також може бути надана можливість вибору часу, який може включати години, хвилини та секунди. Цей елемент інтерфейсу широко використовується в програмах та веб-додатках для введення

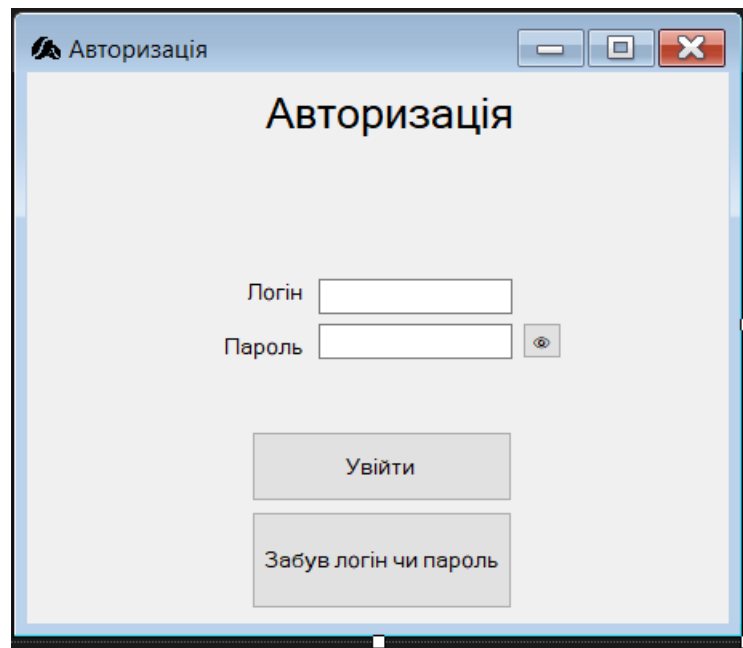
дати та часу в різноманітних сценаріях, таких як заплановані події, налаштування засобів сповіщення, введення строків та інших варіантів, де необхідно вказати точний момент у часі;

- *DataGridView*: Таблиця даних це інтерактивний елемент інтерфейсу, який використовується для відображення та редагування даних у вигляді табличної структури. Елемент надає можливість відображати дані у вигляді рядків і стовпців, що дозволяє користувачеві зручно переглядати, фільтрувати та вносити зміни в дані. Таблиця даних може містити різні типи даних у кожній комірці, такі як текст, числа, дати, зображення та інші. Кожен рядок таблиці зазвичай представляє один запис або об'єкт даних, а кожний стовпець - певну характеристику або атрибут цих записів. Крім відображення даних, елемент також може надавати можливість редагування, сортування, фільтрації та пошуку даних. Користувач може вибирати рядки, копіювати та вставляти дані, а також здійснювати інші дії для редагування таблиці. Цей елемент інтерфейсу широко використовується в програмних додатках та веб-сайтах для відображення та управління великими обсягами даних, такими як списки клієнтів, замовлень, продуктів тощо;

Були створені наступні елементи інформаційної системи:

Форма авторизації(рис 3.9) яка створена для захисту системи від третіх осіб, на ній представлена функцію вводу логіну та пароль з можливість показати пароль натиснувши відповідну кнопку, також якщо користувач забув логін чи пароль є можливість відновлення логіну чи пароль за допомогою перходу на форму відновлення через кнопку «Забув логін чи пароль», якщо був ведений логін чи пароль то для преходу в основе меню потрібно натиснути кнопку «Увійти».

При неправильному вводі логіну чи паролі покажиться повідомлення що воно було введено невірно, кількість спроб необмежена.



Авторизація

Логін

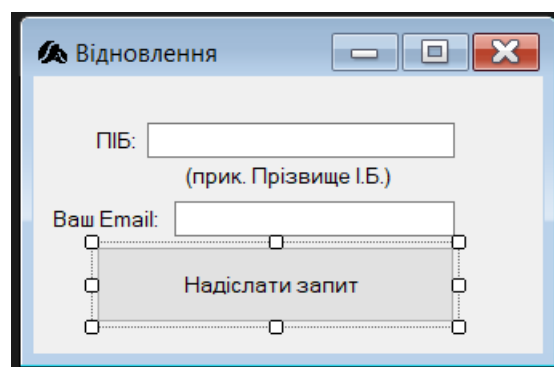
Пароль

Увійти

Забув логін чи пароль

Рисунок 3.9 — вікно авторизації

Форма відновлення логіну чи паролю(рис 3.10) була створена для випадків коли користувач забув данні для входу в програму. На цій формі Потрібно ввести ПІБ користувача та його email. Повідомлення надсилається тех персоналу та тех персонал потім відсилає на вказаний email логін та пароль користувача якщо ПІБ користувача був знайдений в базі даних користувачів.



Відновлення

ПІБ:
(прик. Прізвище І.Б.)

Ваш Email:

Надіслати запит

Рисунок 3.10 — вікно відновлення логіну чи паролю

Код надіслання запиту:

```

private void button1_Click(object sender, EventArgs e)
{
    var smtpClient = new SmtpClient("smtp.gmail.com", 587)
    {
        Credentials = new NetworkCredential("lokarorer@gmail.com", "я
вирізає свій ключ для безпеки"),
        EnableSsl = true
    };
    var mailMessage = new MailMessage
    {
        From = new MailAddress("lokarorer@gmail.com"),
        Subject = "Відновлення логіну та пароллю",
        Body = $"Я {textBox1.Text} забув свій логін та пароль, прошу
надіслати їх на адресу: {textBox2.Text}",
        IsBodyHtml = false
    };
    mailMessage.To.Add("cafehelpcenter@gmail.com")
    try
    {
        smtpClient.Send(mailMessage);
        MessageBox.Show("Електронний лист успішно надісланий!");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка при відправленні електронного
листа: {ex.Message}");
    }
}

```

Форма «Меню»(рис 3.11) була створена для навігації по основній частині програми.

З цієї форми працівники можуть перейти на їх основні форми наприклад звичайний працівник може перейти на форму касира, також з форми можна перейти на форму адміністрації та подивитися розклад роботи працівників, також через меню можна знайти допомогу по касі або допомогу адміністратору, також через меню можна зайти на форму автору де можна подивитись хто створив програму та версію програми.

Якщо в програму ввійшов звичайний працівник то для нього буде недоступними кнопка адміністрації та вибір допомоги адміністратору.

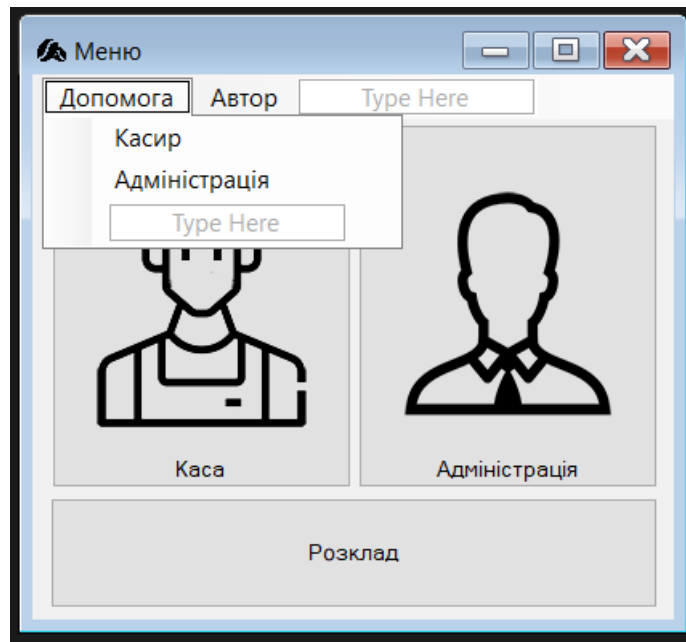


Рисунок 3.11 — вікно меню

На формі «автор»(рис 3.12) можна подивитись хто створив програму та версію програми.

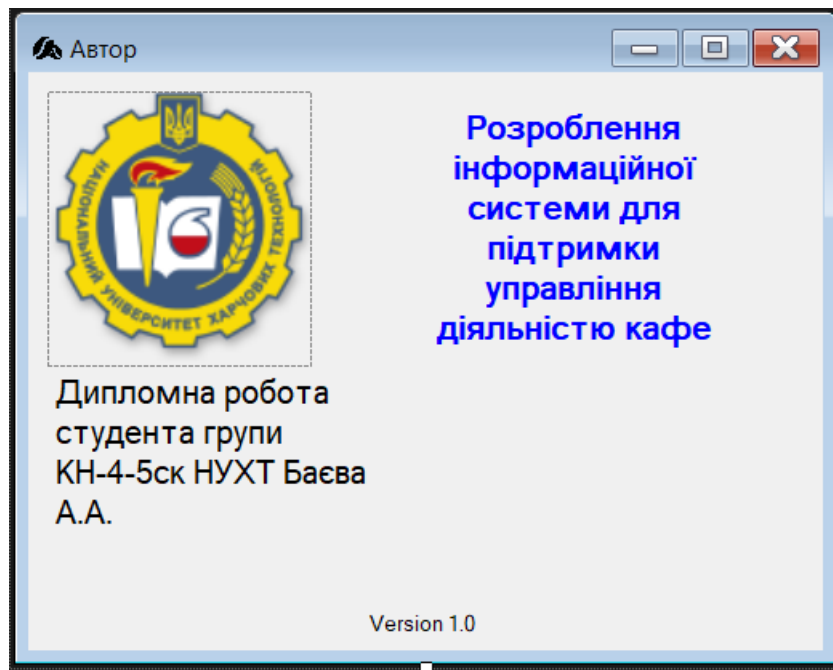


Рисунок 3.12 — Вікно автору

Форма «Каса» (рис 3.13) була створена для Створення, редагування та оформлення замовлень в кафе. До неї має доступ всі працівники кафе, Форма складається з панелей для легшої навігації по формі та для певних функцій.

На панелі меню автоматично створюються кнопки продукції, при кожному заході на форму буде створено повне меню кафе.

Код панелі меню:

```
private void CreateProductButtons()
{
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT ProductID, ProductName FROM
Production";
        SqlCommand command = new SqlCommand(query, connection);
        SqlDataReader reader = command.ExecuteReader();
        int buttonSize = 100;
        int padding = 10;
```

```

int x = padding;
int y = padding;
int panelWidth = Menu.Width;
while (reader.Read())
{
    Button productButton = new Button
    {
        Text = reader["ProductName"].ToString(),
        Size = new Size(buttonSize, buttonSize),
        Location = new Point(x, y),
        Tag = reader["ProductID"]
    };
    productButton.Click += ProductButton_Click;
    Menu.Controls.Add(productButton);
    productButtons.Add(productButton);
    x += buttonSize + padding;
    if (x + buttonSize > panelWidth)
    {
        x = padding;
        y += buttonSize + padding;
    }
}
connection.Close();
}
}

```

В панелі категорії автоматично створюються кнопки типу продукції, при кожному заході на форму буде створено всі типи продукції.

На панелі замовлення відображався номер замовлення та його вміст. При натисненні кнопки з панелі меню в панелі замовлення створюється labels та

buttons що дають можливість подивитись яку продукцію було добавлено та її кількість.

Через кнопку «Оформити замовлення» можна закрити замовлення та оформити чек. Через кнопку «Створити замовлення» можна створити нове замовлення. Також через кнопки типів продукції та строку пошуку можна здійснити фільтр та пошук продукції в панелі меню. На формі можна подивитися поточний час який виставлено в системі та ПІБ працівника який заноситься в чек.

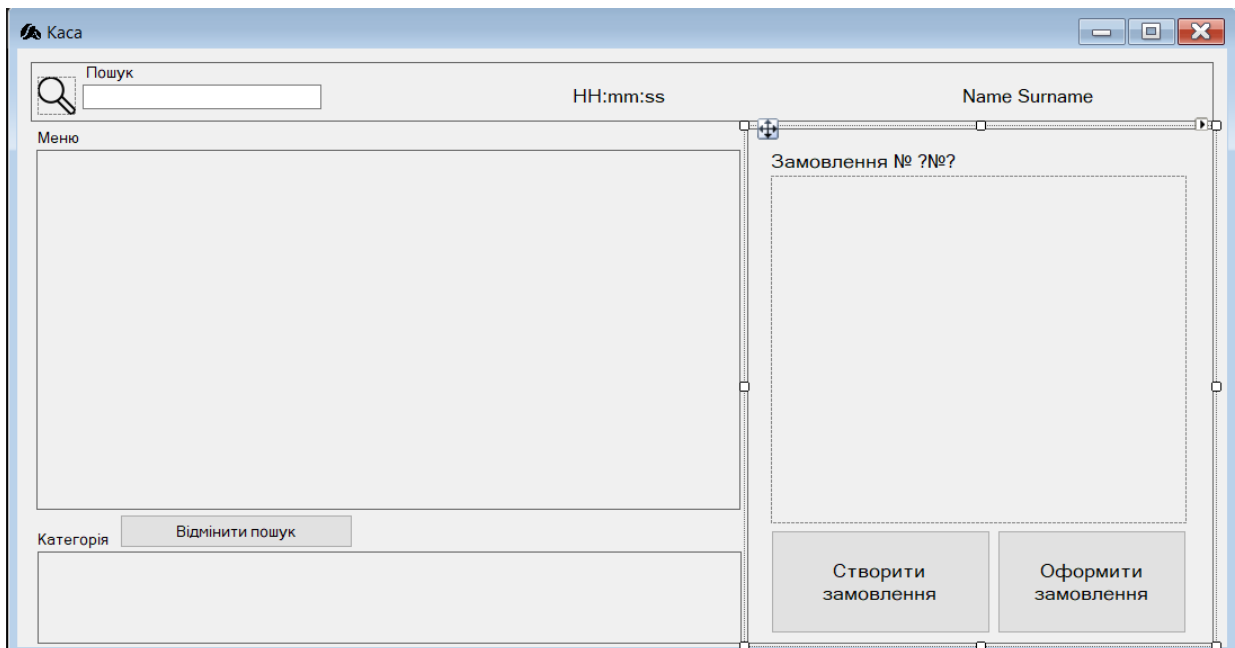


Рисунок 3.13 — Вікно Каса

При натисненні на кнопку «Адміністрація» в формі меню ви перейдете на форму Адміністрації (рис 3.14) в якій присутні елементи кнопок які доступні тільки адміністрації, також тут є меню з довідками для адміністрації.

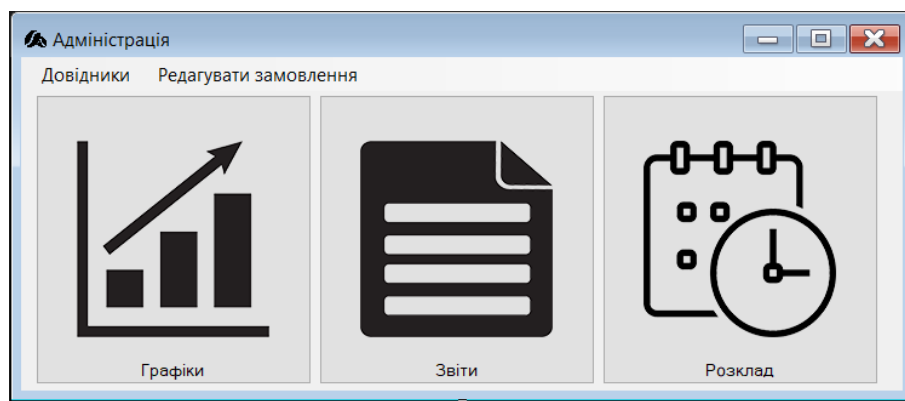


Рисунок 3.14 — Вікно адміністрації

В довідниках(рис 3.15) можна подивитись та редагувати інформації по таблицям.

(Прим. Таблиця працівники)

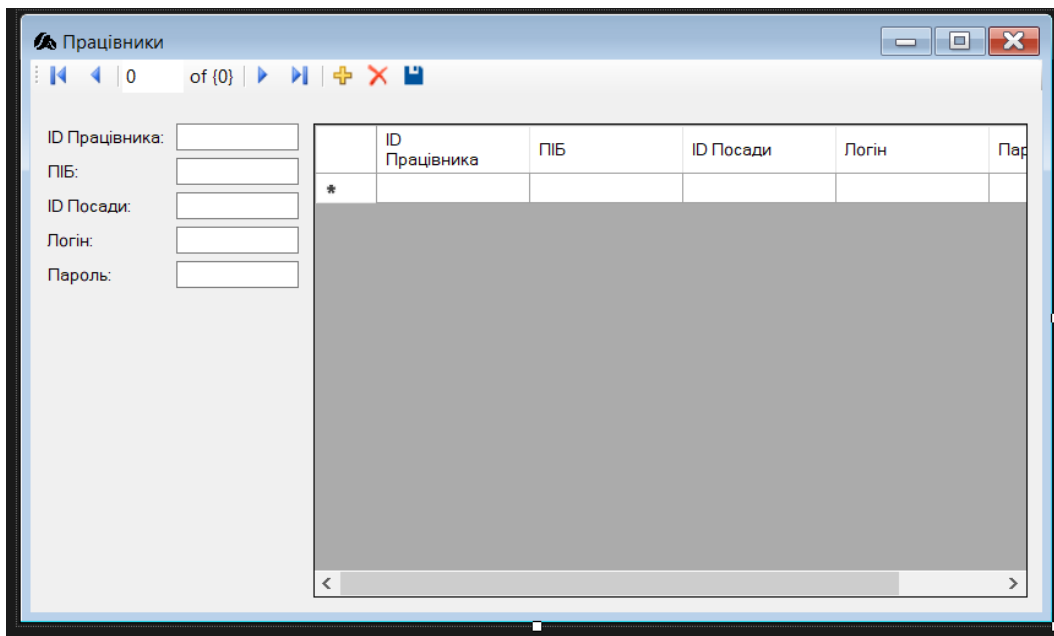


Рисунок 3.15 — вікно працівники

Через меню можна потрапити на форму редагування замовлення(рис 3.16) для адміністрації, на цій формі адмін може знайти любе замовлення по даті або його ІД, також він може вручну створювати та редагувати вміст замовлення використовуючи кнопки та listbox

При виборі замовлення можна сформувати звіт замовлення де буде показано вміст замовлення яке можна роздрукувати

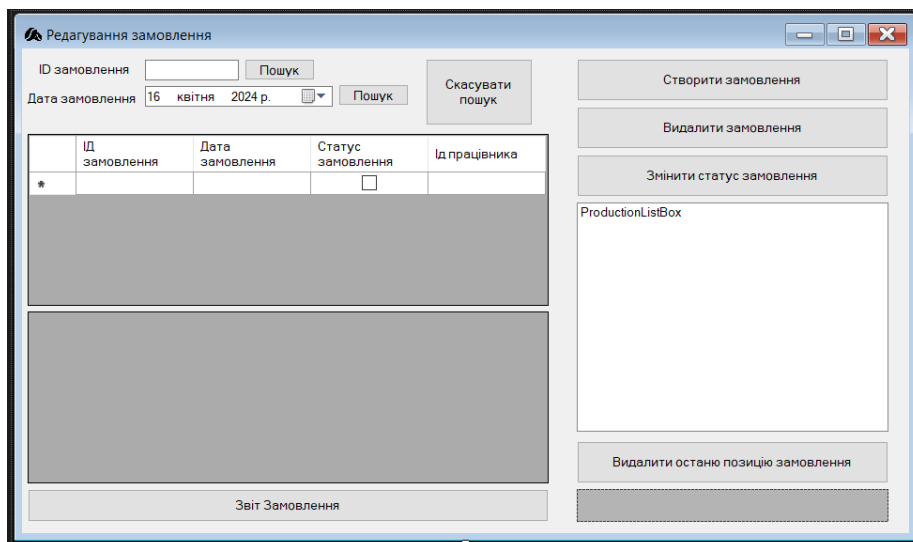


Рисунок 3.16 — Вікно редагування замовлення

З форми адміністрації можна потрапити на форму графіків(рис 3.17) де можна подивитись: Кількість проданої продукції, Сума продажу за місяць, Створено замовлень(Працівником).

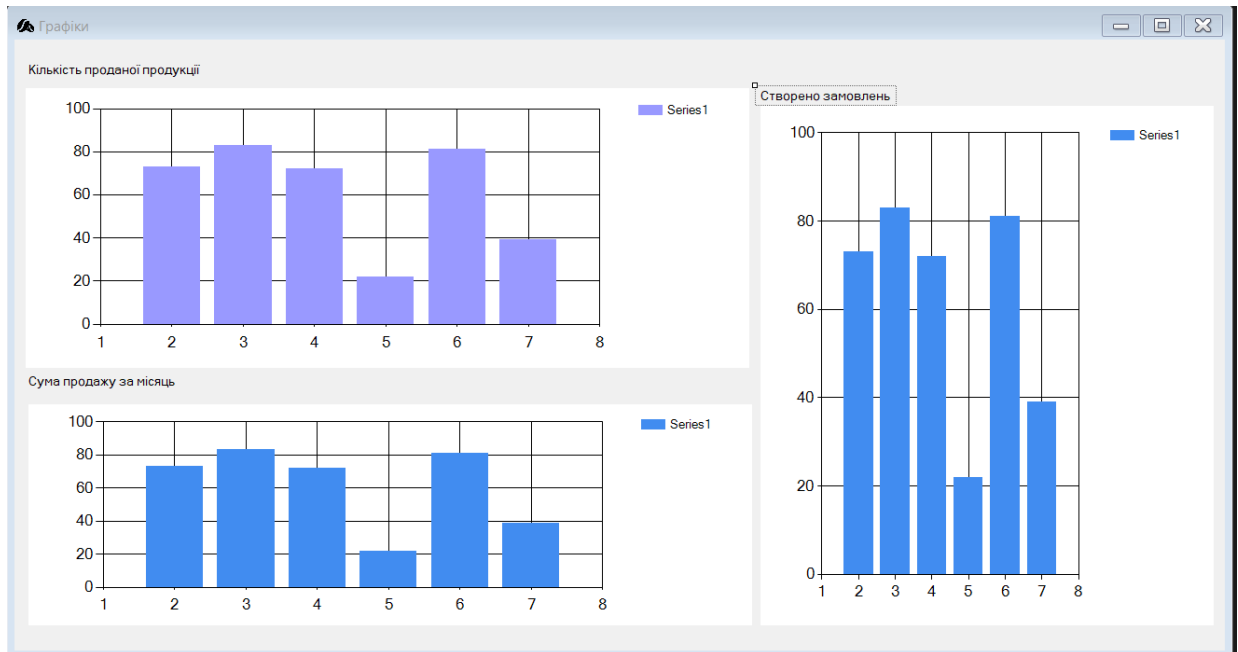


Рисунок 3.17 — Вікно Графіків

Код заповнення одного з графіків:

```

chart1.Series.Clear();
using (SqlConnection connection = new
SqlConnection(connectionString))
{
    string query = "SELECT p.ProductName, SUM(od.Amount) AS
TotalAmount " +
        "FROM Production p " +
        "INNER JOIN [Order Details] od ON p.ProductID =
od.ProductID " +
        "GROUP BY p.ProductName";
    SqlCommand command = new SqlCommand(query, connection);
    connection.Open();
    SqlDataReader reader = command.ExecuteReader();
    while (reader.Read())

```

```

    {
        string productName = reader["ProductName"].ToString();
        double totalAmount =
Convert.ToDouble(reader["TotalAmount"]);
        chart1.Series.Add(productName);
        chart1.Series[productName].Points.AddY(totalAmount);
    }
}

```

Через форму адміністрації при натисненні на кнопку «Звіти» користувач потрапляє на форму Звітів(рис 3.18). На даний момент на цій формі наявні звіти всіх замовлень, звіт працівників та звіт по постачальнику інші звіти будуть доповнюватись в майбутньому за потреби.

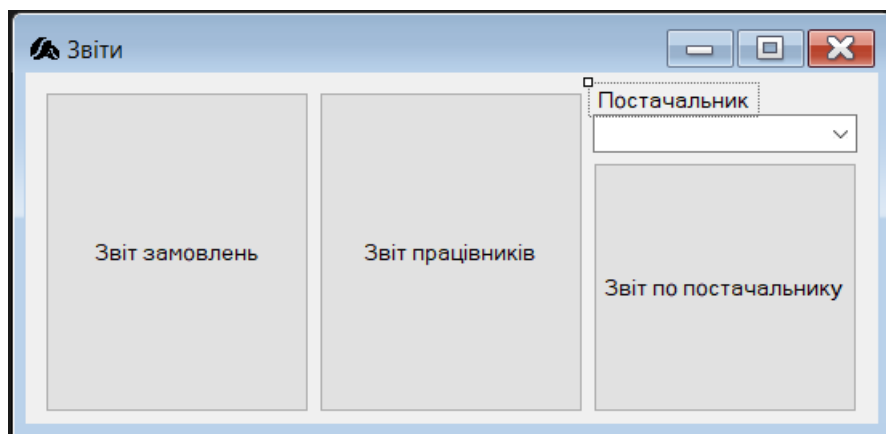


Рисунок 3.18 — Вікно Звіти

Остання кнопка на формі адміністрації можна потрапити на форму розкладу(рис 3.19), адміністрація та тех-персонал закладу має доступ до редагування розкладу. Всі працівники в таблицю добавляються автоматично.

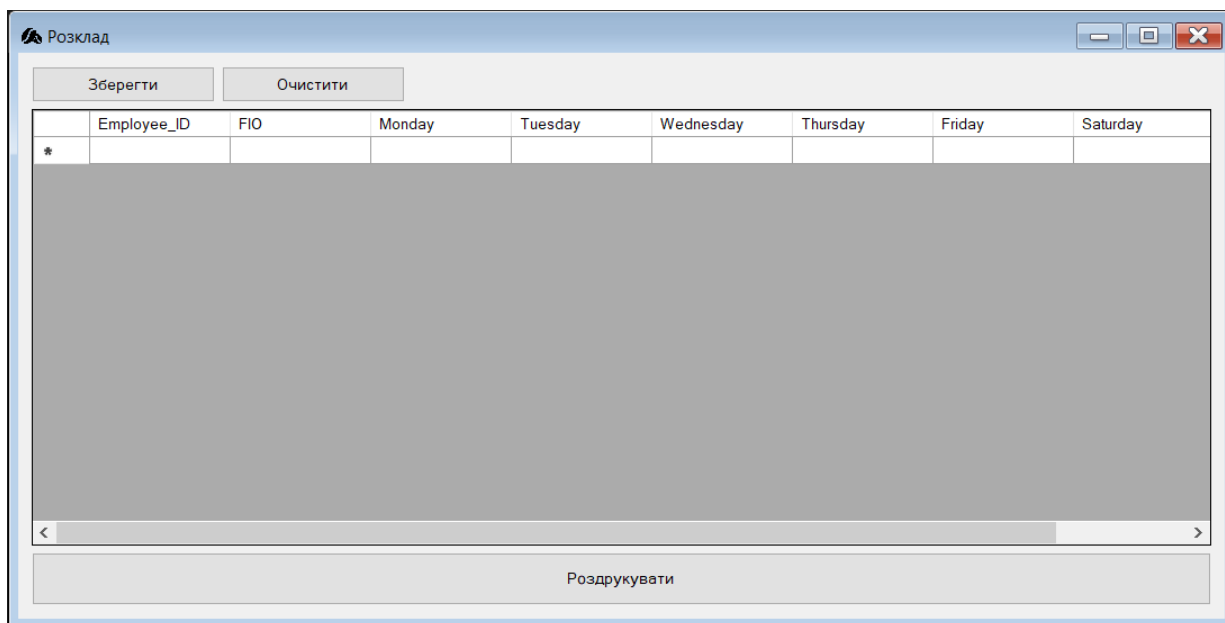


Рисунок 3.19 — Вікно Розкладу

Код таблиці розкладу:

```

public partial class Schedule : Form
{
    private string connectionString = "Data Source=DESKTOP-
Q4GQHJN\\SQLEXPRESS;Initial Catalog=Cafe;Integrated Security=True";
    private Dictionary<string, string> UkrainianDays = new Dictionary<string,
string>()
    {
        {"Monday", "Понеділок"},
        {"Tuesday", "Вівторок"},
        {"Wednesday", "Середа"},
        {"Thursday", "Четвер"},
        {"Friday", "П'ятниця"},
        {"Saturday", "Субота"},
        {"Sunday", "Неділя"}
    };
    public Schedule()
    {
        InitializeComponent();
    }
}

```

```

        this.Load += new EventHandler(Schedule_Load);
    }
    private void Schedule_Load(object sender, EventArgs e)
    {
        // TODO: This line of code loads data into the 'cafeDataSet.Schedule'
table. You can move, or remove it, as needed.
        this.scheduleTableAdapter.Fill(this.cafeDataSet.Schedule);
        CopyEmployeesToSchedule();
        int positionID = (int)AppState.PositionID;
        if (positionID == 2)
        {
            dataGridView1.Enabled = false;
            button1.Enabled = false;
            button2.Enabled = false;
        }
        else
        {
            dataGridView1.Enabled = true;
            button1.Enabled = true;
            button2.Enabled = true;
        }
        TranslateColumnHeaders();
    }
    private void TranslateColumnHeaders()
    {
        foreach (DataGridViewColumn column in dataGridView1.Columns)
        {
            if (UkrainianDays.ContainsKey(column.HeaderText))
            {
                column.HeaderText = UkrainianDays[column.HeaderText];
            }
        }
    }
}

```

```

    }
}
}
private void CopyEmployeesToSchedule()
{
    // SQL-запит для вставки даних з Employees в Schedule
    string query = @"
INSERT INTO Schedule (Employee_ID, FIO)
SELECT Employee_ID, FIO
FROM Employees
WHERE Employee_ID NOT IN (SELECT Employee_ID FROM
Schedule)";
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        connection.Open();
        SqlCommand command = new SqlCommand(query, connection);

        int rowsAffected = command.ExecuteNonQuery();
    }
}
private void button1_Click(object sender, EventArgs e)
{
    this.Validate();
    this.scheduleBindingSource.EndEdit();
    this.scheduleTableAdapter.Update(this.cafeDataSet);
}
private void button2_Click(object sender, EventArgs e)
{
    ClearWeekDays();
}

```

```

}
private void ClearWeekDays(object employeeID = null)
{
    string query = @"
UPDATE Schedule
SET Monday = NULL,
    Tuesday = NULL,
    Wednesday = NULL,
    Thursday = NULL,
    Friday = NULL,
    Saturday = NULL,
    Sunday = NULL";

    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        connection.Open();
        SqlCommand command = new SqlCommand(query, connection);
        command.ExecuteNonQuery(); // Виконання запиту
    }
    this.Validate();
    this.scheduleBindingSource.EndEdit();
    this.scheduleTableAdapter.Update(this.cafeDataSet);
    this.scheduleTableAdapter.Fill(this.cafeDataSet.Schedule);
    this.scheduleTableAdapter.Update(this.cafeDataSet);
}
private void button3_Click(object sender, EventArgs e)
{
    var dgvPrinter = new DGVPrinter();
    dgvPrinter.CreateReport("Розклад", dataGridView1);
} }

```

3.3. Інструкція користувача

Для запуску програми відкриваємо файл «CafeProgram.exe» двічі натиснувши на нього мишею. З'явиться вікно авторизації програми(рис 3.20).

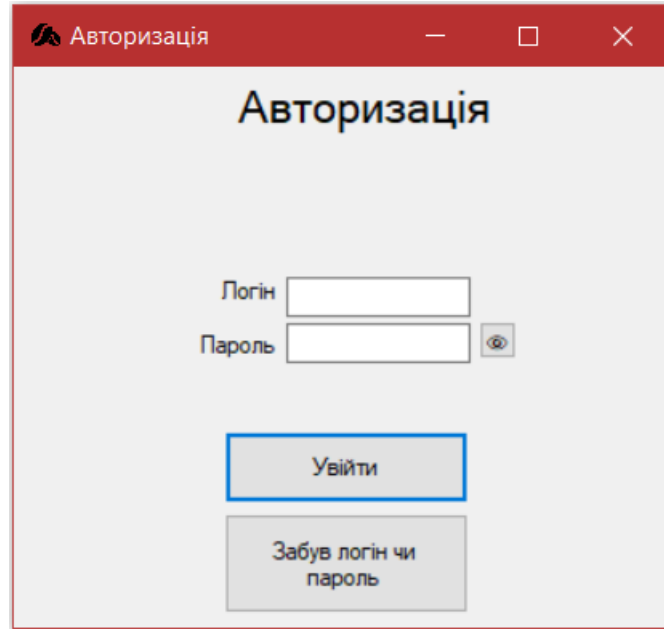


Рисунок 3.20 — Вікно авторизації

На ньому потрібно ввести логін чи пароль та натиснути на кнопку «Увійти» щоб ввійти в програму для цього потрібно навестись на поле вводу, клацнути мишею та ввести логін після цього повторити таке з паролем та навестись мишею та один раз клацнути на кнопку «Увійти» в наступних поясненнях процес введення інформації та натиснення на кнопки буде ідентичним тому він не буде пояснюватись, для зручності при натисненні на око можна показати пароль(за замовчуванням він прихований), якщо логін чи пароль буде вказано невірно то спливе повідомлення(рис 3.21).

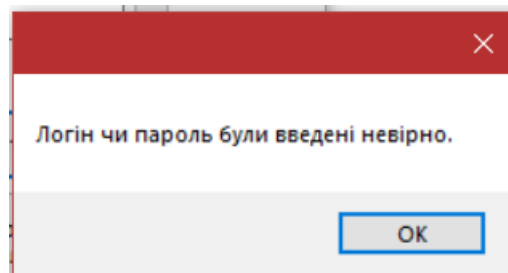


Рисунок 3.21 — Повідомлення неправильно введених даних

При правильному вводі логіну та паролю користувач переміститься на вікно основного меню програми. Якщо користувач забув пароль то він має натиснути на кнопку «Забув логін чи пароль».

При переході на вікно відновлення логіну чи паролю(рис 3.22) потрібно ввести свій ПІБ та назву свого електронного кошику(далі email), після чого натиснути кнопку «Надіслати запит»(рис 3.23)

Рисунок 3.22 — Вікно відновлення логіну чи паролю

Рисунок 3.23 — Повідомлення успішного надіслання листу в техпідтримку

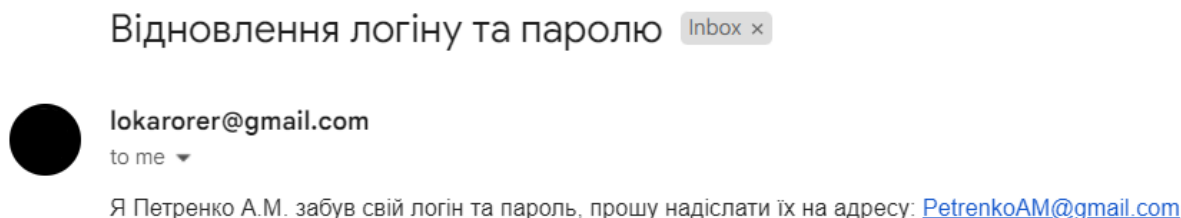


Рисунок 3.24 — Лист який приходить техпідтримці

Техпідтримка перевіре чи є ваш ПІБ в базі даних та в випадку якщо він є надішле ваш логін та пароль на вказаний email(рис 3.24).

Після успішної авторизації користувач потрапляє на вікно основного меню програми(рис 3.25) в залежності від того яка посада у працівника йому буде доступні різні функції системи(рис 3.26).

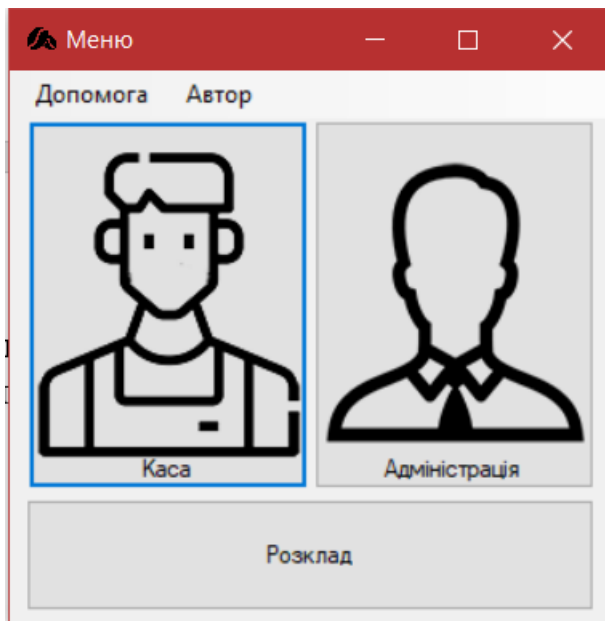


Рисунок 3.25 — вікно меню Адміністратора

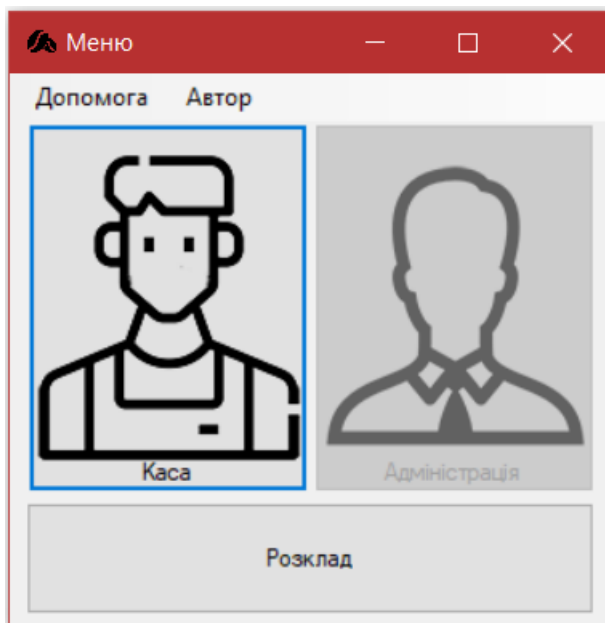


Рисунок 3.26 — вікно меню касира

Далі буде розписано інструкцію для різних посад а саме звичайного працівника та адміністратора так як функції адміністратора та техперсоналу ідентичні

Інструкція Працівника

Для працівника на вікні меню буде доступне вікно каси, розкладу та елементи допомоги «каса» та елемент меню «автор»

При натисненні на елемент меню розкривається його піделементи(рис 3.27)

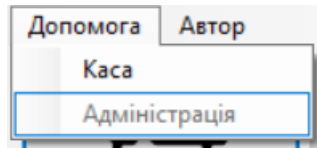


Рисунок 3.27 — елементи меню «допомога»

При натисненні на піделемент «каса» відриється вікно з допомогою(рис 3.28) для працівників по віні каса, при натисненні на елемент «Автор» відкриється вікно автора програми(рис 3.29) де можна подивитись хто створив програму та її версію.

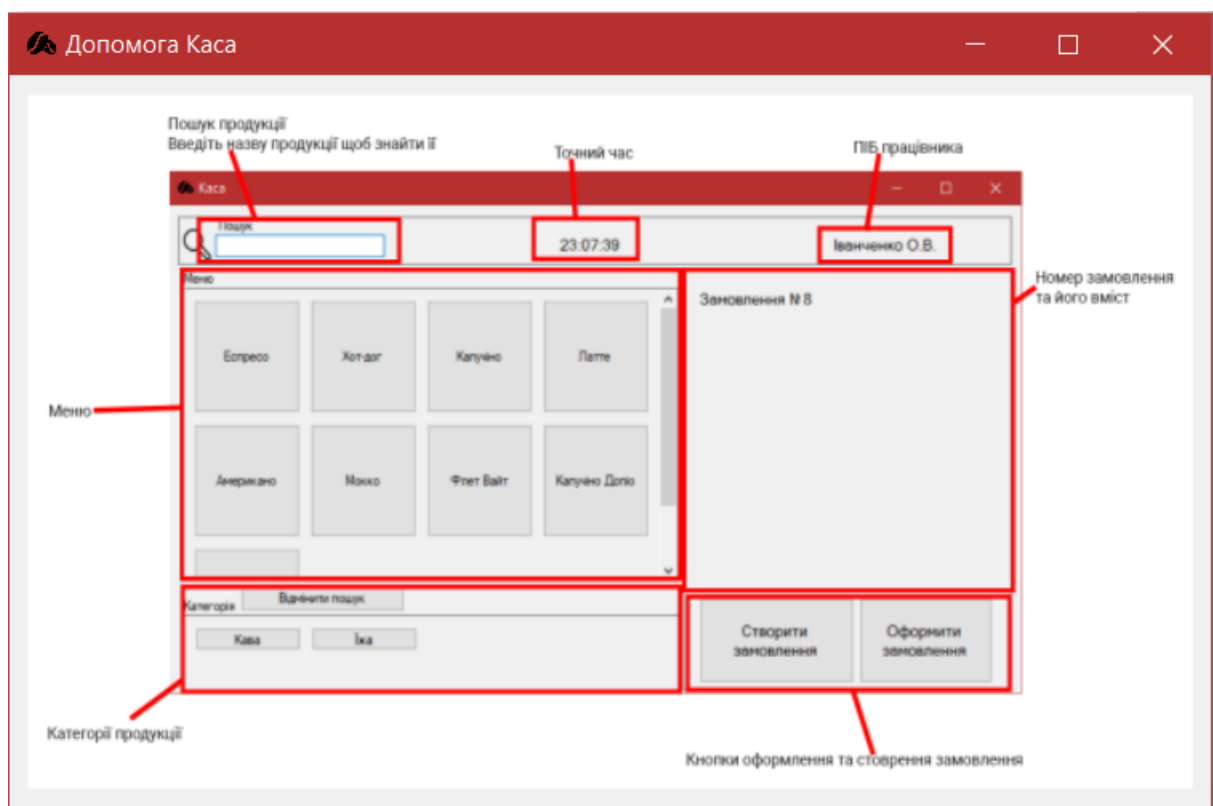


Рисунок 3.28 — Вікно «Допомога каса»



Рисунок 3.29 — Вікно «Автор»

При натисненні на кнопку «Каса» відкриється вікно «Каса» (рис 3.30) де працівник й відтворює основні функції каси.

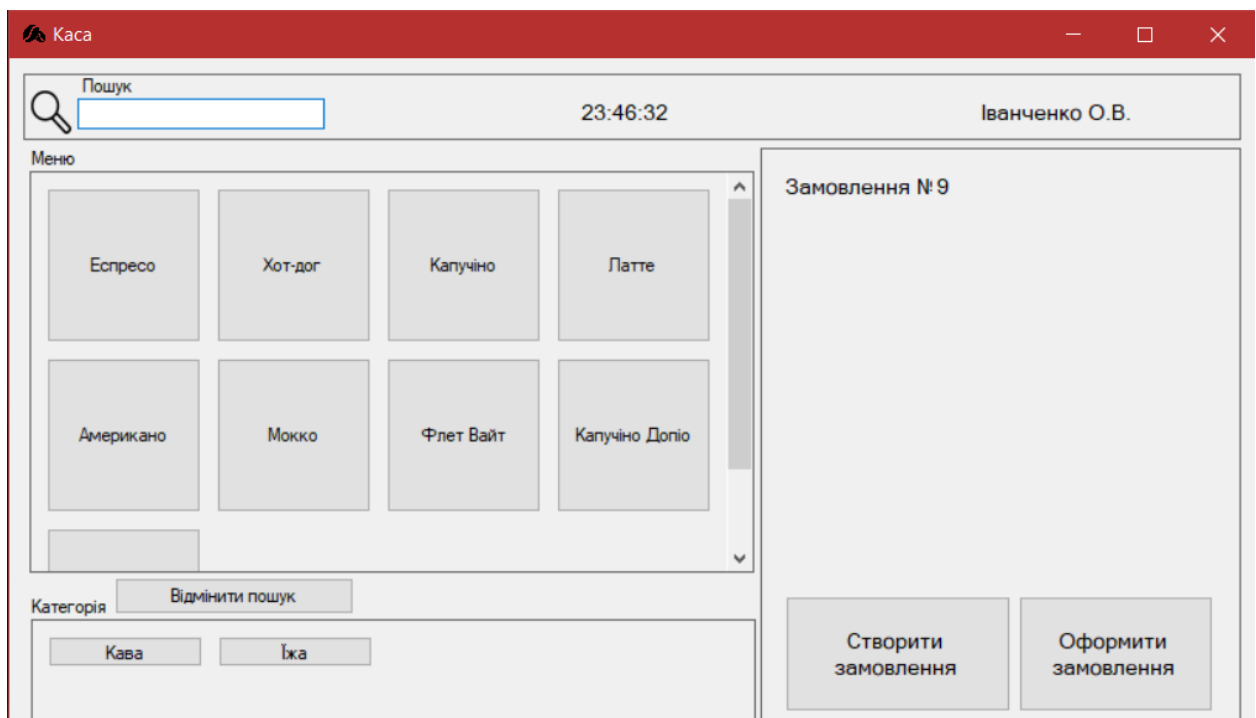


Рисунок 3.30 — Вікно «Каса»

На цьому вікні можна створювати, редагувати, та видаляти замовлення при відкритті вікна замовлення створюється автоматично, щоб додати в замовлення записи потрібно натиснути на кнопку бажаної продукції в меню після чого вона додається до замовлення (рис 3.31).

Замовлення № 9

1	Флет Вайт	-	+	X
1	Мокко	-	+	X

< _____ >

Створити замовлення Оформити замовлення

Рисунок 3.31 — Додані позиції до замовлення

Щоб додати кількість вибраної продукції натисніть кнопку з позначкою плюс(+) навпроти позиції, щоб відняти кількість вибраної продукції натисніть кнопку з позначкою мінус(—) навпроти позиції якщо кількість буде менше 1 одиниці товару то позиція видалиться з замовлення, щоб видалити позицію з замовлення натисніть кнопку з позначкою хрестика(X). Після кінця редагування замовлення потрібно натиснути кнопку «Оформити замовлення» після чого буде відкрито чек(рис 3.32) який потрібно буде роздрукувати. Після друку чеку для обслуговування наступного клієнту потрібно створити нове замовлення натиснувши кнопку «Створити замовлення».

ООО «Кав'ярня»
Чек № 9
Касир: Іванченко О.В.

1. Флет Вайт 1 95
2. Мокко 1 85

Каса: 2
НДС 20%
Дата 2024-05-19 23:55:52

Підсумок = 180 грн

Дякую за покупку!

Рисунок 3.32 — Створений чек

Якщо потрібно знайти якусь певну продукцію ви можете використати пошук через текстове поле «Пошук», при вводі назви будуть блокуватись непідходящі варіанти продукції що залишить потрібну продукцію добре бачимою(рис 3.33).

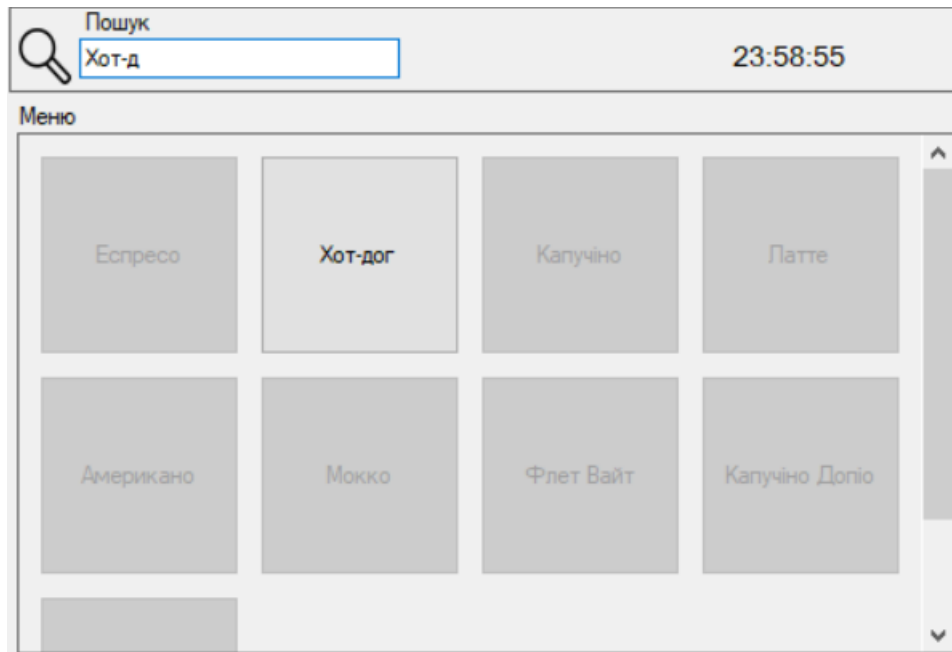


Рисунок 3.33 — Пошук продукції

Також при необхідності можна фільтрувати продукцію за типом через кнопки категорії, для фільтру потрібно натиснути на кнопку з меню категорії(рис 3.34).

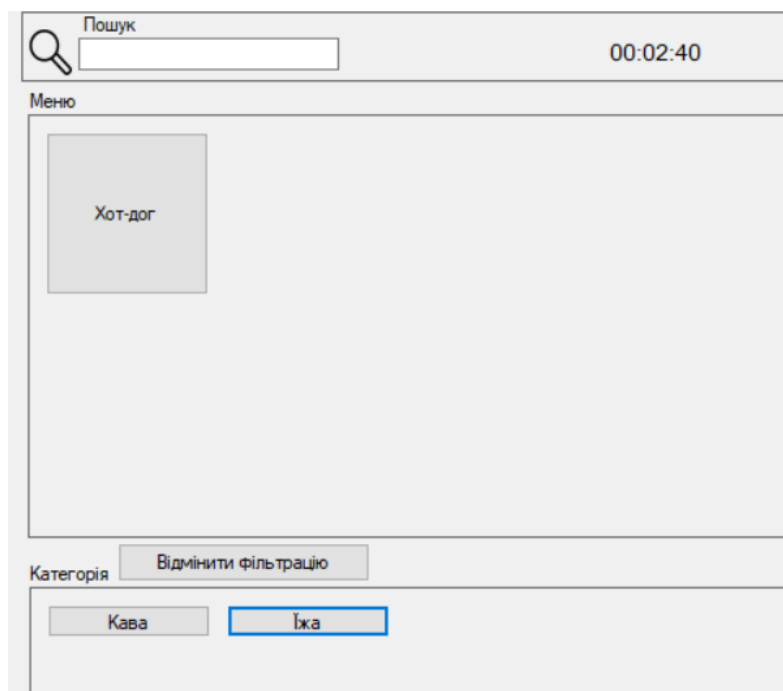


Рисунок 3.34 — Фільтрація продукції

Для того щоб відмінити фільтрацію натисніть кнопку «Відмінити фільтрацію»(рис 3.34).

На цьому функції касира закінчуються. Для того щоб вийти з програми натисніть кнопку хрестик (X) в верхньому правому кутку програми та тисніть на хрестик до моменту форми авторизації чи повного закриття програми.

Інструкція Адміністрації

Для адміністратора доступні всі вищеперераховані функції та нові які будуть вказані нижче.

При натисненні на піделемент меню «Допомога» буде відкрито форму короткої допомоги адміністрації(рис 3.35). При натисненні на кнопку «Адміністрація» буде відкрито вікно меню адміністратора(рис 3.36).



Рисунок 3.35 — Вікно короткої допомоги адміністрації

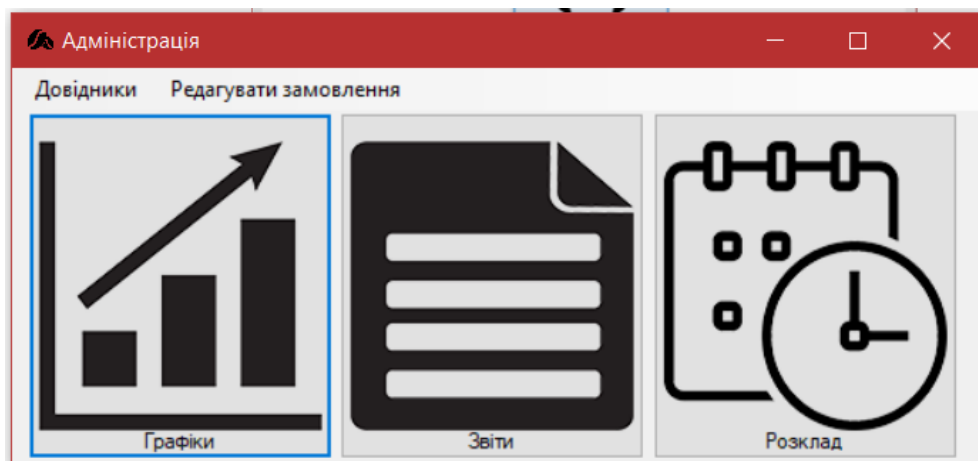


Рисунок 3.36 — Вікно Адміністрація

На цьому вікні є меню в якому є елементи «Довідники» та «Редагувати замовлення» а також кнопки «Графіки», «Звіти», «Розклад».

При натисненні на елемент меню «Довідники» відкриється піделементи

При натисненні на які ви зможете перейти на форми таблиць де ви зможете редагувати таблиці. Вони включають в себе додавання працівників в заклад(рис 3.37), додавання продукції, їх категорій та постачальників, а також при необхідності редагування посад в закладі.

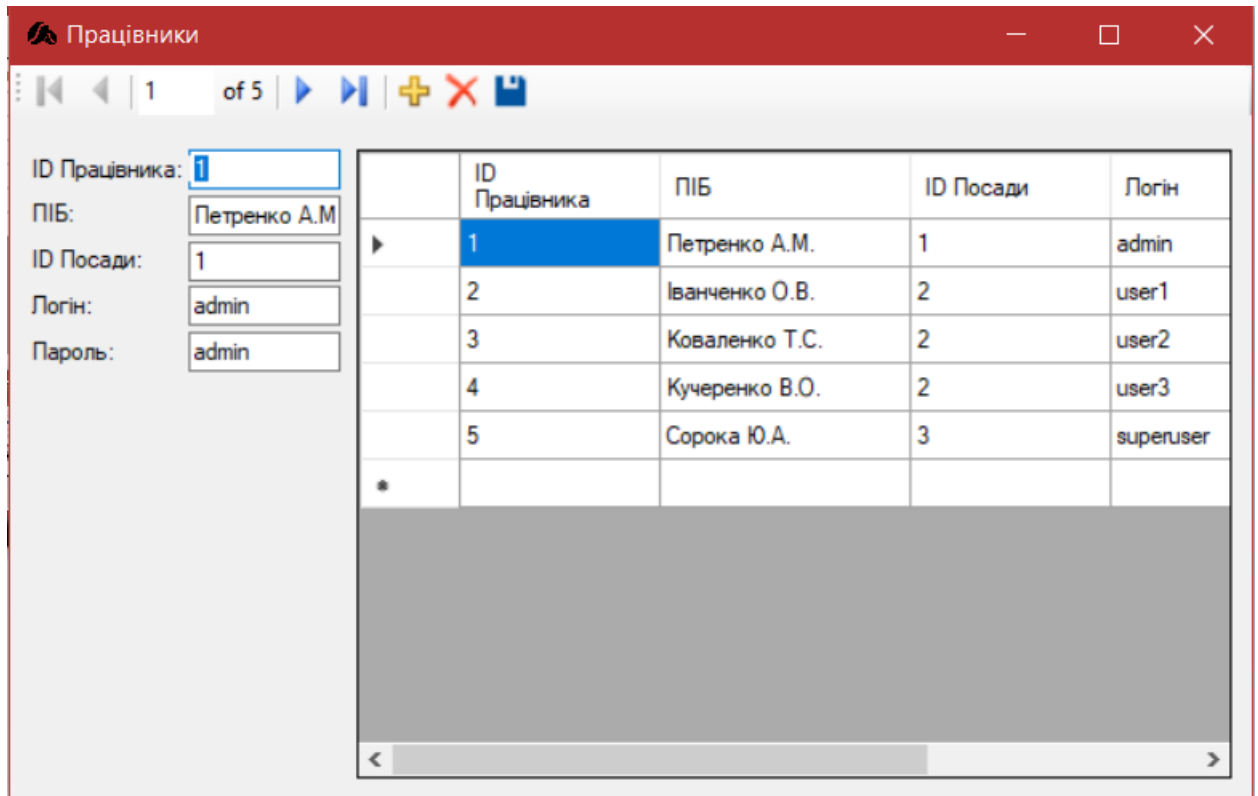


Рисунок 3.37 — Вікно Працівники

На цих довідках є елементи навігації де можна вибрати, додати, редагувати, видалити та зберегти запис.

При натисненні на елемент меню «Редагувати замовлення» відкриється вікно редагування замовлення(рис 3.38) де адміністрація або техпідтримка може зручно знайти замовлення за його ІД та датою(щоб відмінити пошук натисніть кнопку «Скасувати пошук»), створити, видалити та редагувати замовлення та його позиції, а також сформуванати звіт вибраного замовлення.

Для того щоб сформуванати звіт клацніть на замовлення після чого клацніть на кнопку «Звіт замовлення» Після чого відкриється налаштування друку та передпоказ друку звіту(рис 3.39). Для додавання позиції в замовлення

натисніть на потрібне замовлення після чого виберіть позицію що хочете додати в списку продукції, через таблиці ви можете редагувати кількість продукції, а також при повторному натисненні на продукцію в списку буде збільшуватись його кількість, для створення, видалення замовлення, зміни його статусу та видалення останньої позиції з замовлення натисніть на відповідні кнопки.

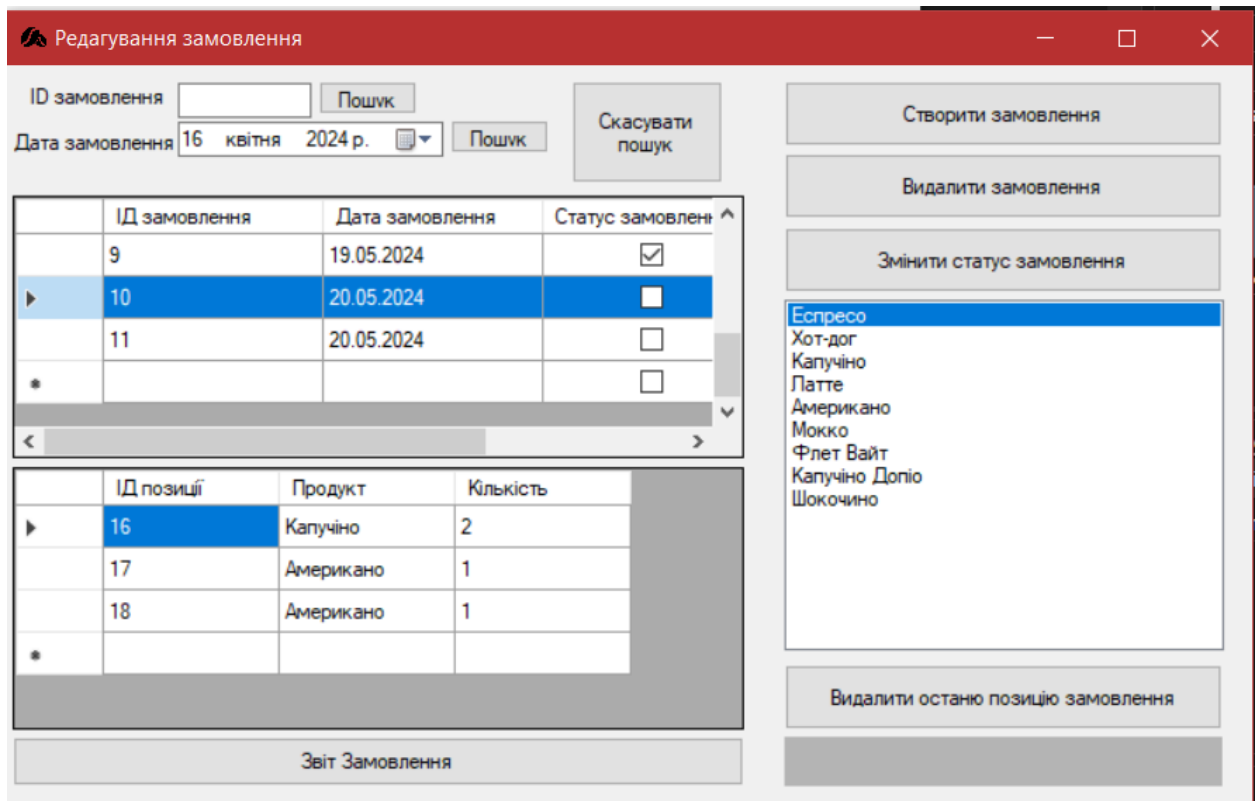


Рисунок 3.38 — Вікно редагування замовлення



Рисунок 3.39 — Вікно звіту замовлення

При натисненні кнопки «Звіти» в вікні адміністрації відкриється вікно звітів(рис 3.40), з меню звітів можна створити звіти натиснувши на кнопку відповідного звіту. Після натиснення на кнопку формування звіту відкриється вікно звіту який можна буде надрукувати(рис 3.41).

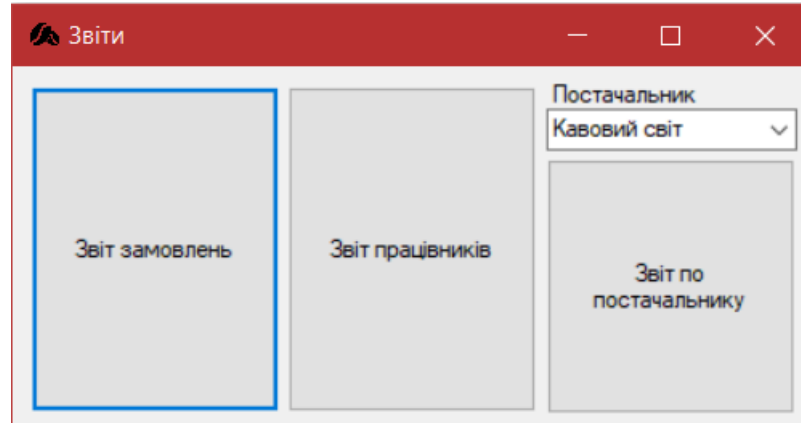


Рисунок 3.40 — Вікно звітів

ID замовлення	Дата замовлення	Статус замовлення	ID працівника
1	25 04 2024	<input checked="" type="checkbox"/>	2
2	14 05 2024	<input checked="" type="checkbox"/>	1
3	14 05 2024	<input checked="" type="checkbox"/>	1
4	14 05 2024	<input checked="" type="checkbox"/>	1
5	14 05 2024	<input checked="" type="checkbox"/>	1
6	19 05 2024	<input checked="" type="checkbox"/>	1
7	19 05 2024	<input checked="" type="checkbox"/>	1
8	19 05 2024	<input checked="" type="checkbox"/>	2
9	19 05 2024	<input checked="" type="checkbox"/>	2
10	20 05 2024	<input checked="" type="checkbox"/>	2
11	20 05 2024	<input checked="" type="checkbox"/>	1

Рисунок 3.41 — Вікно звіту замовлень

При натисненні кнопки «Розклад» відкривається вікно розкладу(рис 3.42). Також на вікно розкладу можна потрапити через головне меню програми.

На вікні розклад автоматично генерується всі ПІБ-и працівників для того щоб можна було записати їх години роботи, записувати їх має доступ тільки адміністратор та техпідтримка, для збереження розкладу потрібно натиснути кнопку «Збереги», для того щоб видалити всі записи годин роботи потрібно

натиснути кнопку «Очистити». В разі необхідності друку розкладу потрібно натиснути кнопку «Роздрукувати» після чого буде створений звіт розкладу який можна буде роздрукувати(рис 3.43).

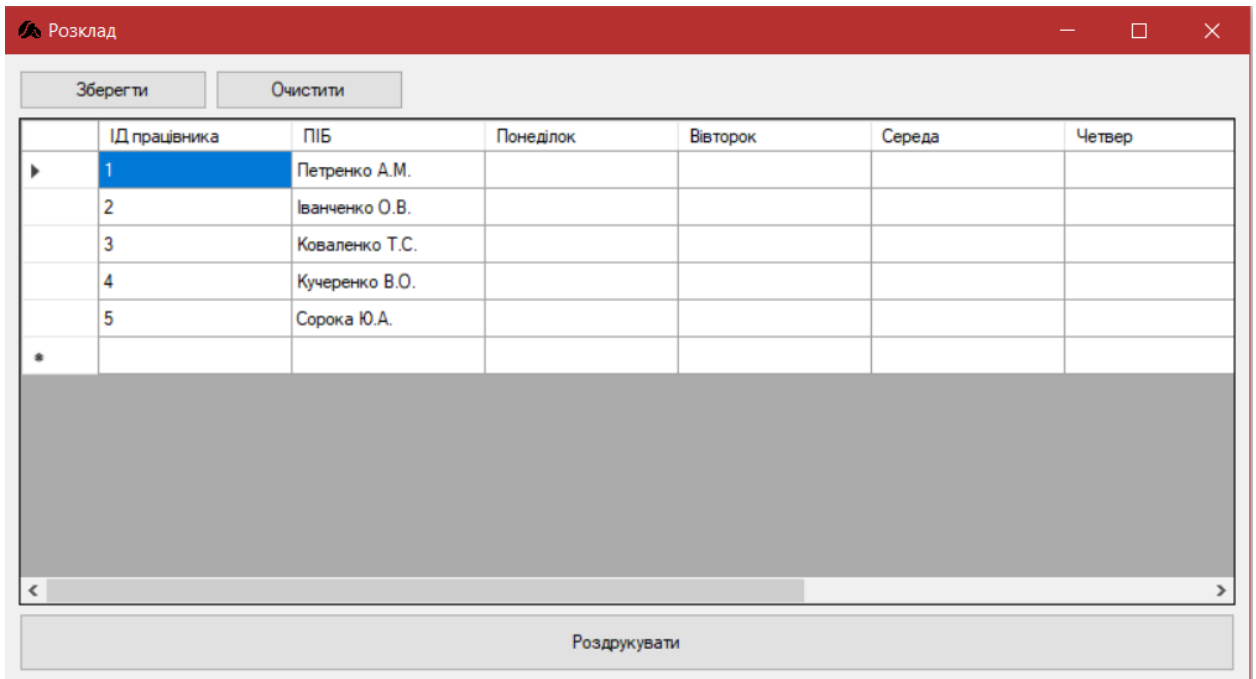


Рисунок 3.42 — Вікно розкладу

Розклад							
ІД працівника	ПІБ	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
1	Петренко А.М.						
2	Іванченко О.В.						
3	Коваленко Т.С.						
4	Кучеренко В.О.						
5	Сорока Ю.А.						

Рисунок 3.43 — Звіт розкладу

Для того щоб вийти з програми потрібно натиснути кнопку з позначкою хрестика(x) в правому верхньому кутку програми до моменту поки не відкриється вікно авторизації або поки програми на закриється повністю.

3.4. Технічне та системне забезпечення розробки

3.4.1. Обґрунтування вибору технічних засобів

Для створення інформаційної системи по темі моєї дипломної роботи було обрано платформу Microsoft Visual Studio 2022. В якості мови реалізації програми обраний C #.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також вебсайти, вебзастосунки, вебслужби як у рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.[3]

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research.

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато від своїх попередників — мов C++, Object Pascal, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова C#, на відміну від C++, не передбачає множинне успадкування класів.

Станом на вересень 2023 року поточна стабільна версія мови C# 11.0, яка була випущена в 2022 році як частина платформи .NET 7.0.[4]

Для розробки функціональної моделі був використаний програмний продукт CA Erwin Process Modeler 7.3 — потужний інструмент для моделювання, створений компанією Computer Associates. Цей інструмент застосовується для аналізу, документування та реорганізації складних бізнес-процесів. Модель, створена за допомогою Erwin Process Modeler, дозволяє детально описати різні аспекти діяльності: дії, які необхідно виконати, способи їх реалізації, необхідні ресурси тощо.

Таким чином, формується цілісна картина діяльності підприємства — від моделей організації роботи в невеликих відділах до докладних ієрархічних структур. Це досягається за допомогою методологій IDEF0 (функціональне моделювання), DFD (моделювання потоку даних) та IDEF3 (моделювання стадій роботи). При розробці або закупівлі програмного забезпечення моделі бізнес-процесів служать чудовим засобом документування потреб, сприяючи забезпеченню високої ефективності інвестицій у сферу ІТ.

AllFusionErwinDataModeler (Erwin) 7.3 - erwin Data Modeler (стилізований як erwin, але раніше як ERwin) — це комп'ютерне програмне забезпечення для моделювання даних. Спочатку розроблений компанією Logic Works, erwin згодом був придбаний низкою компаній, а потім був виділений приватною інвестиційною компанією Parallax Capital Partners, яка придбала та зареєстрована як окрема компанія erwin, Inc., якою керує генеральний директор Адам Фамуларо.

Двигун програмного забезпечення базується на методі IDEF1X, хоча тепер він також підтримує діаграми, що відображаються з нотацією інженерної інформаційної технології, а також нотацією розмірного моделювання.[5]

В якості СУБД було вирішено використати програмний продукт компанії Microsoft SQL Server 2019

Microsoft SQL Server — система управління базами даних, яка розробляється корпорацією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших

застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі.

Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.[6]

3.4.2. Розрахунок та визначення топології комп'ютерної мережі

Лінійна топологія буде простим і ефективним варіантом для такої маленької мережі. Мережа буде містити наступні елементи:

Маршрутизатор (Router): Бездротовий маршрутизатор для забезпечення бездротового підключення комп'ютера(ноутбук) сервера до мережі.

Комп'ютер(ноутбук) сервер та принтери: Один комп'ютер що виконує також роль серверу та два принтери, один pos принтер для чеків та другий для звітів.

Побудова топології:

1. Комп'ютер підключений бездротово до маршрутизатора;
2. Принтери підключаються до комп'ютера через кабелі Ethernet;

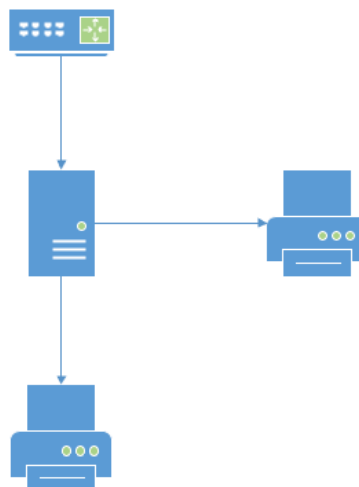


Рисунок 3.4.2 — Схема мережі

Лінійна топологія пропонує просте та ефективне розміщення пристроїв в мережі кафе. Вона забезпечує надійне з'єднання між комп'ютером, сервером та принтерами, що дозволяє ефективно використовувати мережеві ресурси.

3.4.3 Обґрунтування вибору ОС та протоколу обміну даними

Вибір операційної системи Windows для проекту автоматизації інформаційної системи кафе має кілька обґрунтованих причин. По-перше, Windows має широке поширення, що робить її зрозумілою для багатьох користувачів. Це спрощує впровадження та навчання персоналу. Крім того, вона має велику кількість програмного забезпечення, яке вже існує на ринку для автоматизації бізнес-процесів, таких як управління замовленнями, облік складу та фінансове планування. Це дозволить кафе легко інтегрувати різні програмні рішення та використовувати їх для підтримки управлінських функцій. Крім того, Windows має велику спільноту користувачів та розробників, що може забезпечити доступність підтримки та додаткових ресурсів для кафе у разі потреби. Обрання лінійної топології для мережі кав'ярні може бути обґрунтоване кількома факторами. По-перше, лінійна топологія є однією з найпростіших і не вимагає складних налаштувань. Кожен пристрій підключений послідовно, що спрощує розгортання мережі та зменшує витрати на обладнання. Для маленької кав'ярні з обмеженим числом пристроїв це ефективний вибір, оскільки відповідає масштабу мережі.

Лінійна топологія також полегшує підтримку та ремонт мережі. У разі виникнення проблеми з одним пристроєм, виявлення і усунення несправностей стає простішим, оскільки пристрої розташовані в послідовному порядку. Крім того, для лінійної топології не потрібно додаткових комутаторів або маршрутизаторів, що дозволяє знизити витрати на обладнання.

Нарешті, лінійна топологія спрощує процес адміністрування. Вона не вимагає складних налаштувань для взаємодії між пристроями, що робить її зручним вибором для кав'ярні з обмеженими людськими та фінансовими ресурсами. Таким чином, обрання лінійної топології для мережі кав'ярні

обґрунтоване з точки зору ефективності, економії та простоти адміністрування.

3.4.4. Розробка і обґрунтування стратегії адміністрування системи

Вимоги полягають у підтримці одного комп'ютера для обробки замовлень та рахунків, зберігання даних та двох принтерів для друку. Обмеження становлять обмежений бюджет та персонал для адміністрування. Тому потрібно налаштування, підтримка та моніторинг сервера та принтерів, а також регулярне резервне копіювання даних.

Буде використано програмне забезпечення для моніторингу, таке як Nagios або Zabbix, а також вбудовані засоби для резервного копіювання даних Veeam Backup & Replication. Необхідно регулярно проводити моніторинг роботи системи та виконувати резервне копіювання даних на зовнішній накопичувач.

3.4.5. Заходи захисту від несанкціонованого доступу до системи

Встановлено систему авторизації в систему, використовується антивірусне програмне забезпечення.

4 ОХОРОНА ПРАЦІ

Розроблена в даній роботі інформаційна система для автоматизації роботи кафе буде встановлена на персональному комп'ютері з підключенням до мережі інтернет. Крім того, робоче місце користувача інформаційної системи може бути оснащено іншим периферійним обладнанням, таким як термінал для оплати картками, принтер, інша комп'ютерна техніка. В даному розділі будуть розглянуті можливі небезпеки, вплив на роботу шкідливих факторів та заходи щодо їх уникнення задля безпечної та комфортної роботи користувачів інформаційної системи.

Використання комп'ютерної техніки, в тому числі і персонального комп'ютера, зовсім не таке безпечне, як можна подумати. Це переконливо підтверджується інформацією, що міститься у стандарті ГОСТ 12.0.003-2015, який описує основні шкідливі та небезпечні фактори такого трудового процесу. До їх списку входять:

- Значний рівень зорових навантажень, що зазнають працівником;
- Ризик ураження статичною електрикою у разі випадкового контакту з окремими елементами техніки;
- Недостатня освітленість робочої зони;
- Підвищений рівень напруженості електричного та магнітного полів, що генеруються працюючою технікою;
- Високий рівень блискучості та контрастності робочого екрану, що негативно впливає на зір працівника у тривалій перспективі;
- Інші фактори;

Щоб донести до працівника вимоги, які встановлює техніка безпеки під час використання комп'ютера, роботодавець зобов'язаний організувати інструктаж. Він проводиться на підставі спеціально складеної інструкції, яку зазвичай розробляє спеціаліст з охорони праці або безпосередній керівник працівника. Проведення інструктажу необхідно організувати на початок

самостійної роботи: фахівець допускається до роботи лише після його проходження.

Це означає, що до нього донесено правила техніки безпеки при роботі з комп'ютером. Інструкція-пам'ятка з техніки безпеки за комп'ютером має бути видана кожному працівнику у паперовому чи електронному вигляді. Їй можна дати вигляд інфографіки, що легко читається – наприклад, 10 правил техніки безпеки за комп'ютером. Заходи безпеки перед початком роботи. У рамках проведення інструктажу до працівника доводяться відомості про необхідні дії, що виконуються на кожному етапі робочого процесу. Техніка безпеки на початку роботи на комп'ютері вимагає виконання наступних операцій:

- Перевірити справність елементів електросистеми, що забезпечує живлення комп'ютера, включаючи електропроводку, вимикачі, вилки та розетки, за допомогою яких апаратура підключається до мережі;
- Проконтролювати заземлення комп'ютера; перевірити його працездатність.

При виконанні робіт потрібно дотримуватись наступних правил:

- Забороняється класти на корпус та дисплей комп'ютера сторонні предмети;
- У разі виявлення несправності комп'ютера негайно припинити роботу та повідомити про це безпосереднього керівника; Експлуатувати комп'ютер лише за дотриманням інструкції, встановленої виробником;
- Уникати частого та необґрунтованого увімкнення та вимкнення комп'ютера під час роботи.

Після завершення роботи співробітнику необхідно виконати такі дії:

- Вимкнути комп'ютер із використанням алгоритму, встановленого виробником;
- Знеструмити периферійне обладнання;
- Переконавшись у відключенні техніки;
- Виконати очищення робочих поверхонь вологою тканиною.

Вимоги до розташування працівника за комп'ютером націлені на забезпечення його комфорту протягом усієї робочої зміни та відсутність негативних наслідків тривалої роботи. Вони діють для будь-яких робочих місць, чи то бухгалтерія, навчальний клас чи кабінет інформатики. Вони включають такі правила:

- Повна опора ступнями на підлогу під час посадки;
- Використання комп'ютерних меблів, що відповідають нормам СанПіН 2.2.2/2.4.1340-03;
- Відмова від схрещування кінцівок, здатного утруднити кровообіг;
- Дотримання відстані до монітора комп'ютера не менше 45 сантиметрів;
- Правильне встановлення освітлення, яке не повинно світити в очі та залишати відблиски на робочому моніторі.

Для полегшення тривалої роботи експерти рекомендують використовувати допоміжне обладнання, наприклад, особливі окуляри, антиблікові монітори, килимки для миші та інше. Порушення техніки безпеки під час роботи на комп'ютері здатні викликати стійкі розлади здоров'я, які потім важко буде ліквідувати.

Тривалість безперервної праці та відпочинку для працівників, які постійно працюють за персональною офісною технікою, регламентована на законодавчому рівні. Відповідні норми наводяться у СанПіН 2.2.2/2.4.1340-03, який встановлює обов'язкову тривалість перерв у такій роботі. Вони залежать від довжини робочої зміни співробітника та визначаються так:

- При 8-годинній робочій зміні залежно від тяжкості та напруженості праці працівника загальна тривалість перерв за робочий день має становити від 50 до 90 хвилин;
- При 12-годинній робочій зміні – від 80 до 140 хвилин.

Проводити такі перерви за монітором, витрачаючи їх на читання новин або онлайн-ігри, не можна. Працівнику потрібно присвятити їх фізичним вправам, короткій прогулянці, гімнастиці для очей та іншій активності.

Тривалість зазначених перерв згідно з чинним Трудовим кодексом включається до загальної тривалості робочого часу: робочий день працівника не продовжується за рахунок таких періодів відпочинку. Також у цей час працівник не може залучатись до іншої роботи. Крім цих перерв, йому надається стандартний час для обіду [7].

ВИСНОВОК

Метою бакалаврської роботи була розроблення інформаційної системи для підтримки управління діяльністю кафе.

Для вирішення даної задачі було досліджено та детально проаналізовано діяльність кафе, досліджено, які потоки даних є в таких закладах та операції, що виконують працівники. Для цього створено функціональну модель предметної області та виявлено функції що підлягають автоматизації.

Для збереження та аналізу даних в середовищі SQL server було створено базу даних інформаційно-пошукової системи. Інтерфейс додатку був реалізований за допомогою мови програмування C# в середовищі Visual Studio 2022.

Програма призначена для автоматизації обліку замовлень в кафе, що підвищить ефективність роботи персоналу та оптимізацію діяльності з метою збільшення прибутку. Крім того, це забезпечення надійного збереження інформації, швидкого пошуку та формування звітної документації.

Створена комп'ютерно-інформаційна система надає можливість створювати, обробляти та видаляти замовлення та вести облік проданої продукції, редагувати ціну на продукцію та створювати чеки замовлень, меню продукції. В підсумку створена система автоматизує багато облікових функцій кафе, що сприятиме підвищенню ефективності та конкурентоспроможності діяльності кафе.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Історія ресторанів з самої їх появи. — Електронний ресурс – Режим доступу: <https://presa.com.ua/navchannia/istoriya-restoraniv-z-samoji-jikh-poyavi.html>
2. Порівняння SADT - DFD методологій структурного моделювання. — Електронний ресурс – Режим доступу: <http://um.co.ua/8/8-19/8-191285.html>
3. Техніка безпеки під час роботи з персональним комп'ютером. — Електронний ресурс – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
4. C Sharp. — Електронний ресурс – Режим доступу: https://uk.wikipedia.org/wiki/C_Sharp
5. Erwin Data Modeler. — Електронний ресурс – Режим доступу: https://en.wikipedia.org/wiki/Erwin_Data_Modeler
6. Microsoft SQL Server Erwin Data Modeler. — Електронний ресурс – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server
7. Наказ Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду «Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин» від 26.03.2010 №65 – Електронний ресурс. – Режим доступу: <http://zakon2.rada.gov.ua/laws/show/z0293-10>
8. Техніка безпеки під час роботи з персональним комп'ютером. — Електронний ресурс – Режим доступу: <https://pro-op.com.ua/article/485-nstruktsiya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter>
9. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затвержені постановою Головного державного санітарного лікаря України від 10.12.1998 №7 – Електронний ресурс. – Режим доступу: <http://mozdocs.kiev.ua/view.php?id=2445>

10. Джордж Тілманн (2008), «Побудова логічної моделі даних» - Електронний архів. – Режим доступу: <https://web.archive.org/web/20080509063521/http://www.dbmsmag.com/9506d16.html>
11. POS система HOREKER- Електронний ресурс. – Режим доступу: <https://www.horeker.com/>
12. SkyService.pro - хмарна каса для автоматизації бізнесу - Електронний ресурс. – Режим доступу: <https://skyservice.pro/uk/main/>
13. POS Poster- Електронний ресурс. – Режим доступу: <https://joinposter.com/ua>
14. MIT ліцензія – Електронний ресурс – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D1%86%D0%B5%D0%BD%D0%B7%D1%96%D1%8F_MIT

ДОДАТКИ

Додаток А. Моделі бази даних

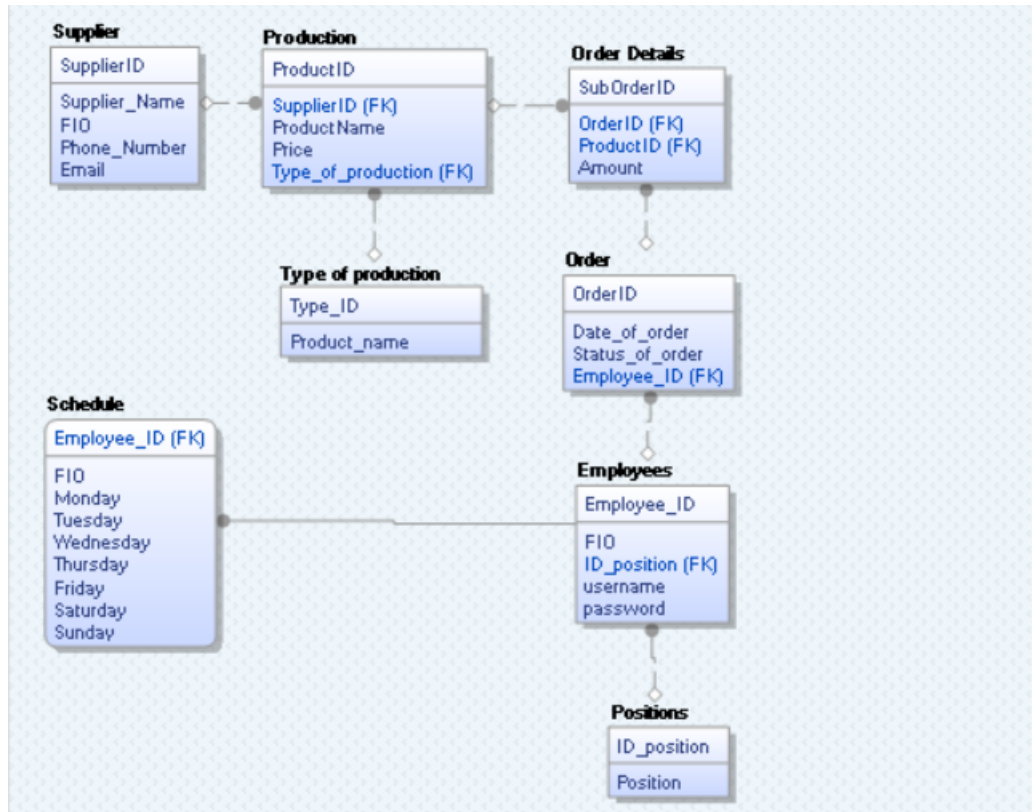


Рисунок А.1. — Логічна модель бази даних

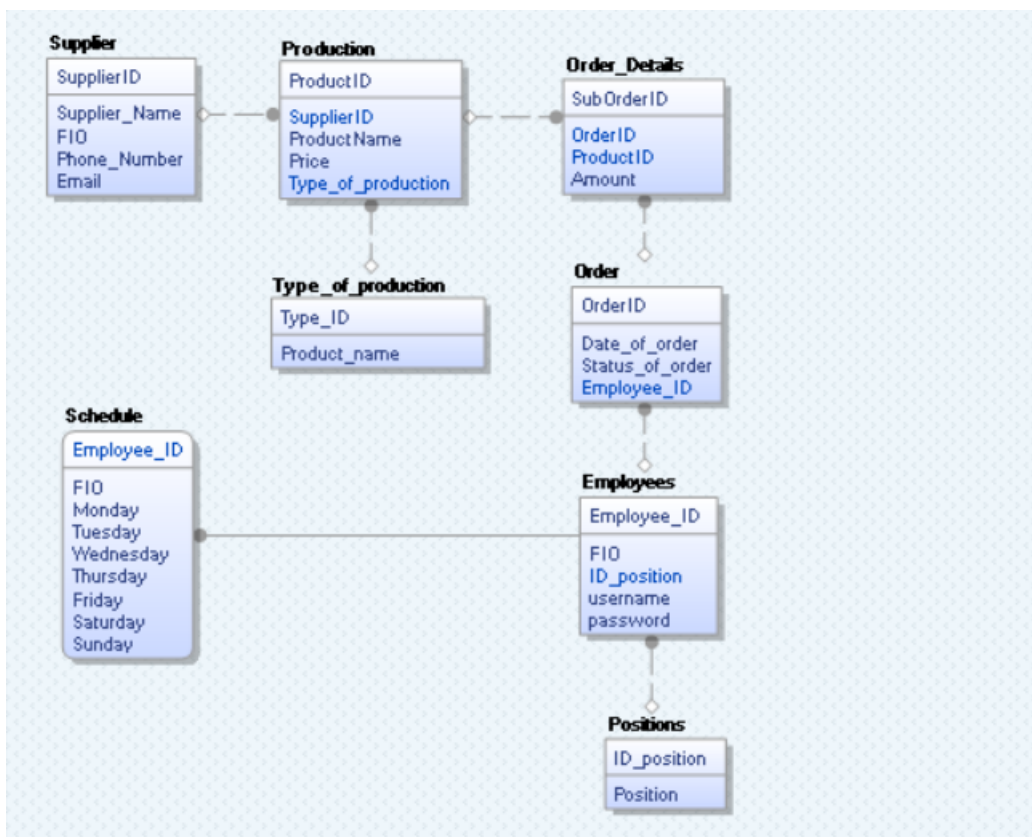


Рисунок А.2. — Фізична модель бази даних

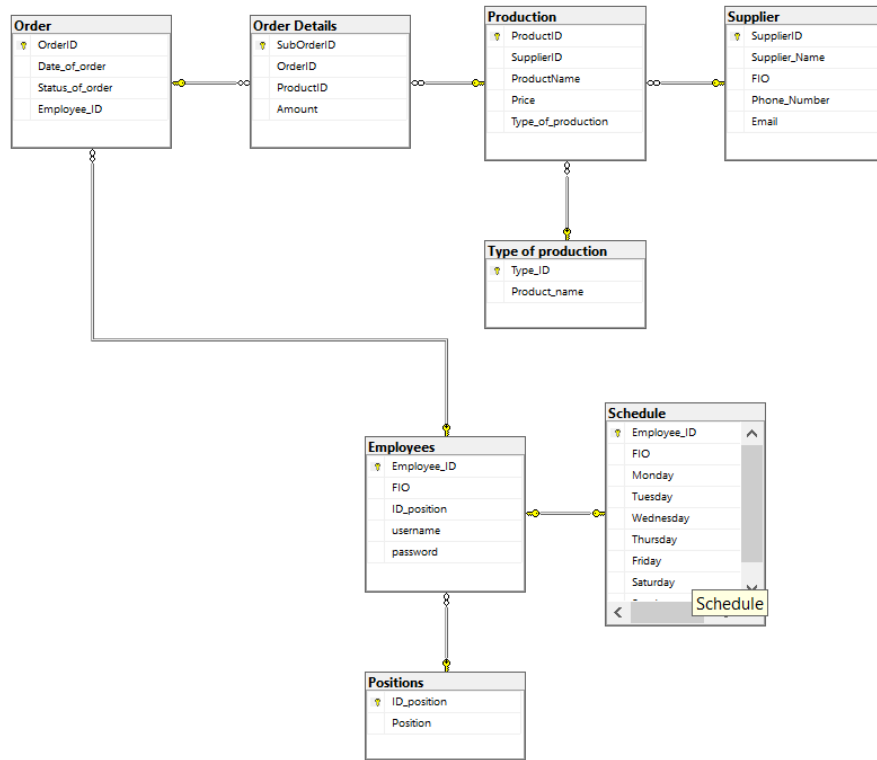


Рисунок А.3. — Діаграма імпортованої бази даних

Додаток Б. Інтерфейс програми

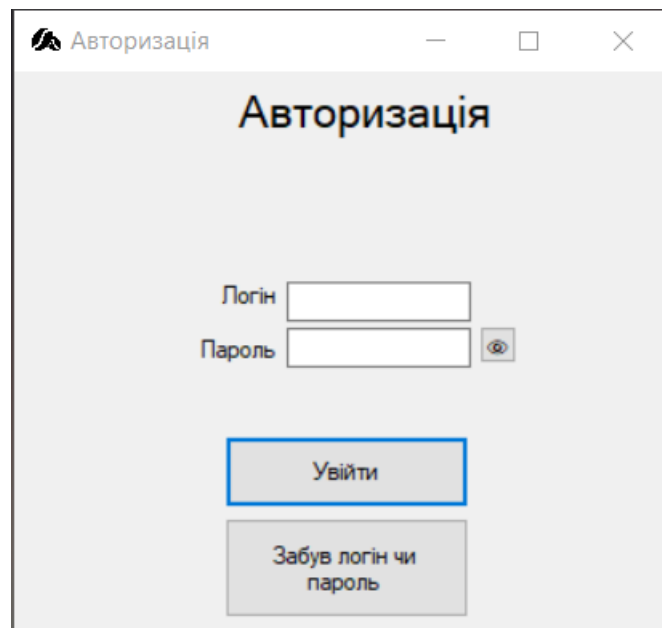


Рисунок Б.1 – Вікно «авторизації»

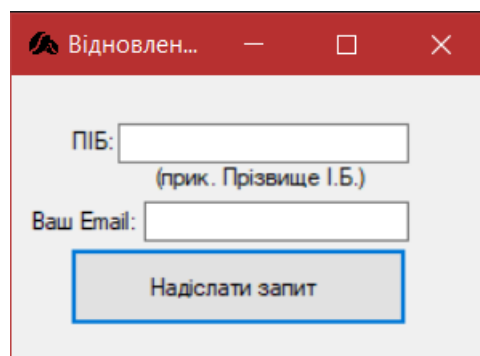


Рисунок Б.2 – Вікно «відновлення логіну та паролю»

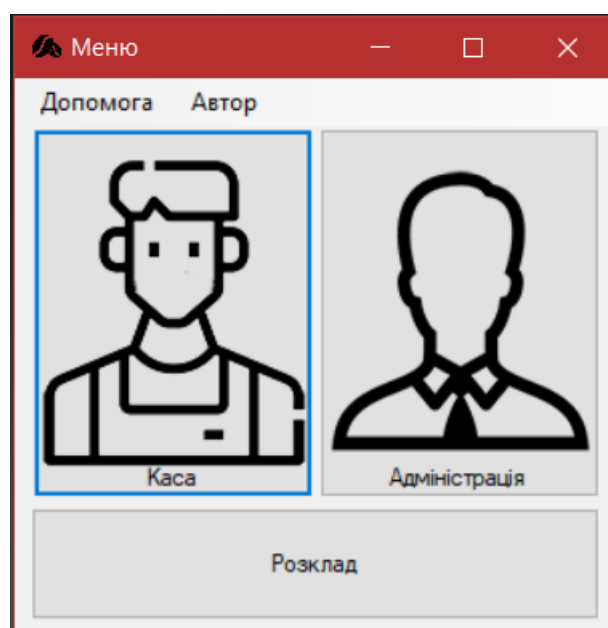


Рисунок Б.3 – Вікно «Меню»

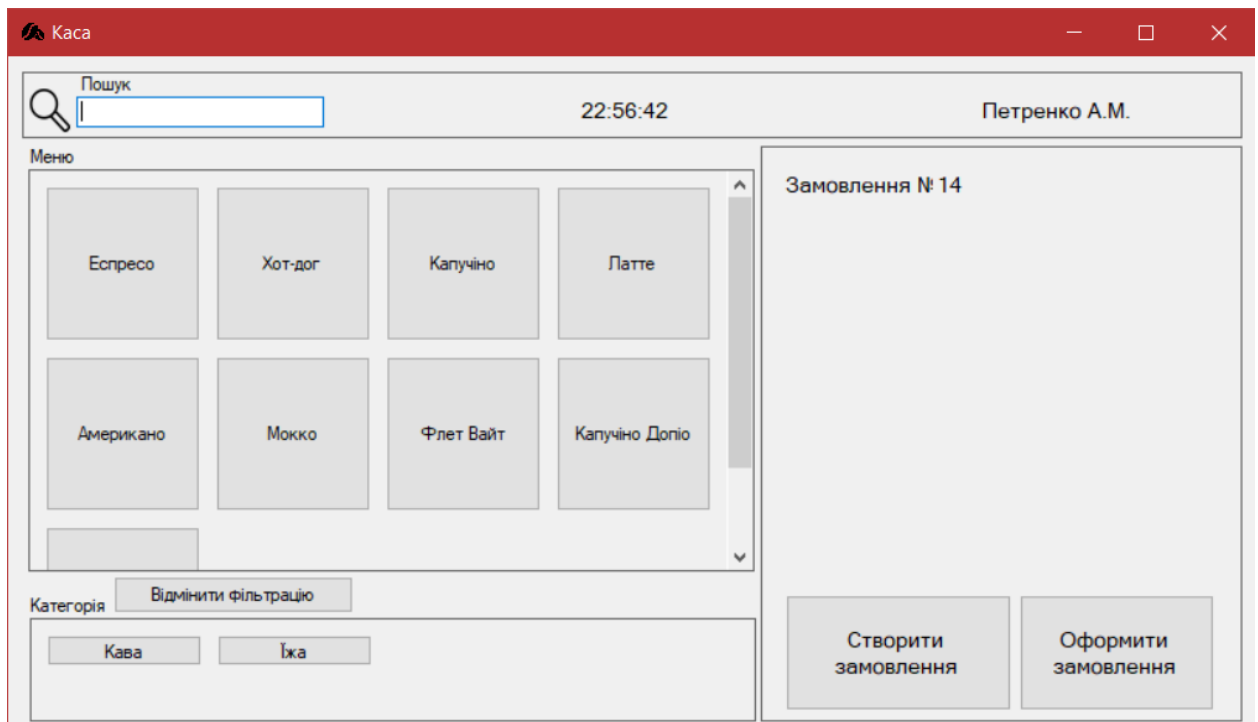


Рисунок Б.4 – Вікно «Каса»

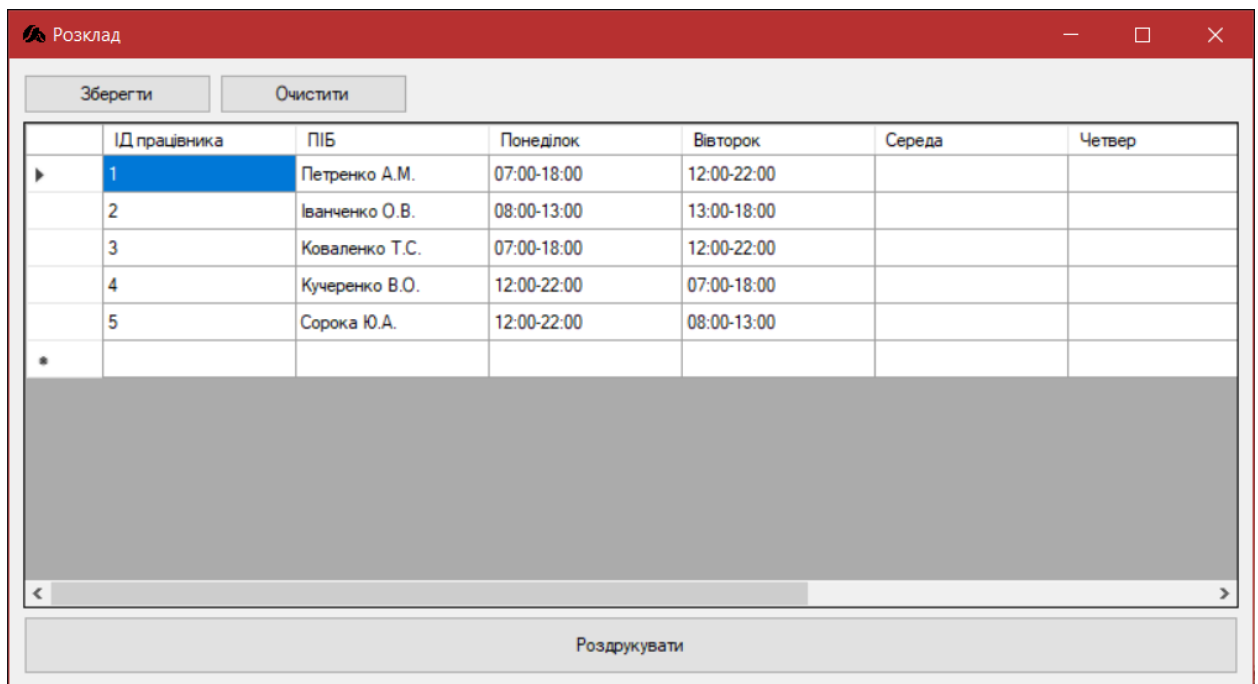


Рисунок Б.5 – Вікно «Розклад»



Рисунок Б.6 – Вікно «Автор»

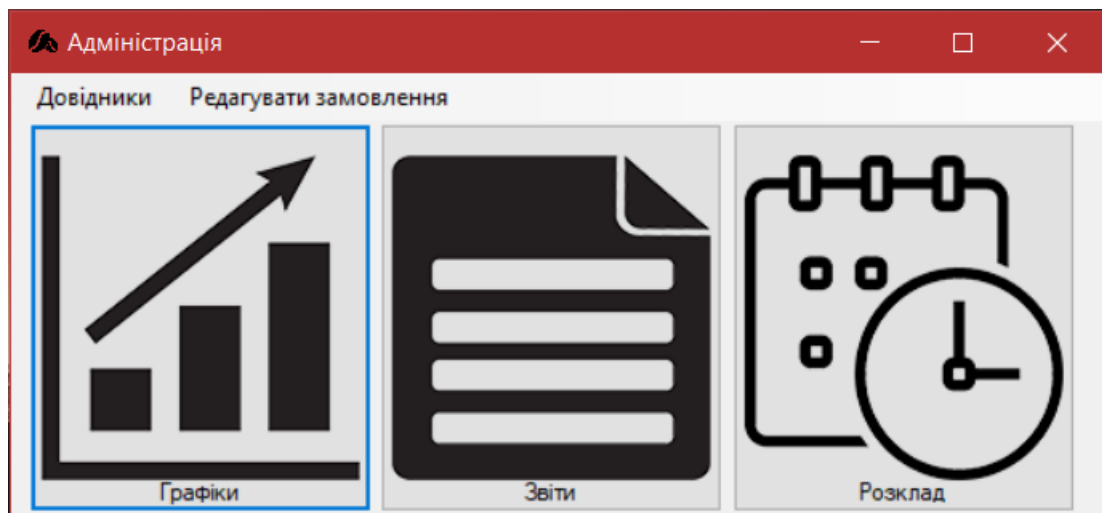


Рисунок Б.7 – Вікно «Авторизація»

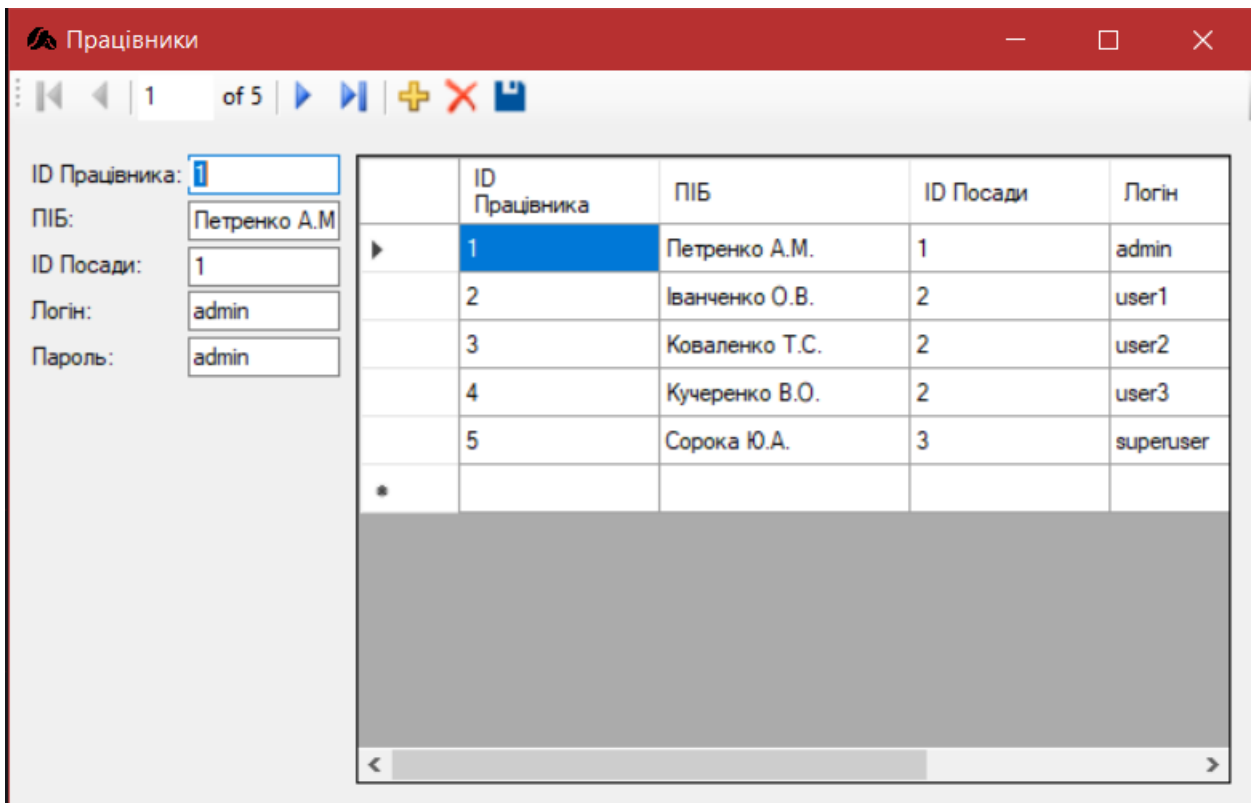


Рисунок Б.8 – Вікно «Працівники»

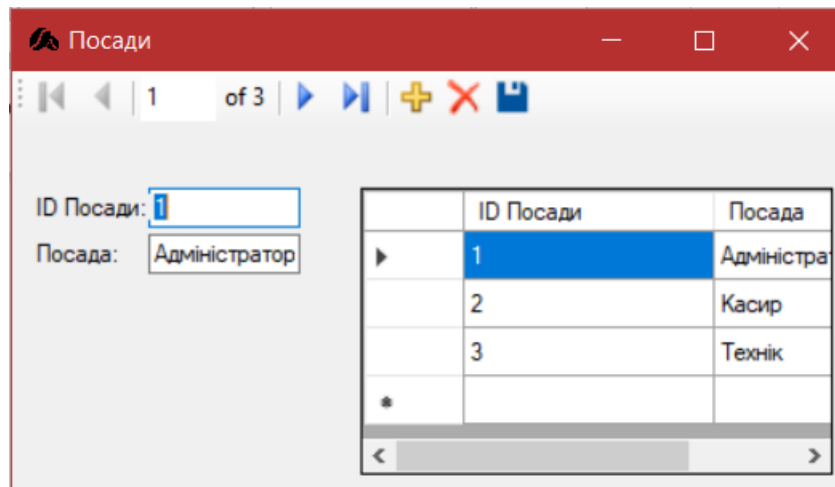


Рисунок Б.9 – Вікно «Посади»

Постачальник

1 of 5

ID Постачальника:

Постачальник: ПІБ:

Номер телефону: Почта:

ID Постачальника	Постачальник	ПІБ	Номер телефону
1	Кавовий світ	Іванов І.І.	+380501234567
2	Ароматна кава	Петров П.П.	+380501234568
3	Елітне кава	Сидоренко С.С.	+380501234569
4	Смачна кава	Михайленко М.М.	+380501234570
5	Кавовий рай	Олександров О.О.	+380501234571

Рисунок Б.10 – Вікно «Постачальник»

Продукція

1 of 9

ID Продукції:

ID Постачальника:

Назва продукту:

Ціна:

Тип продукції:

ID Продукції	ID Постачальника	Назва прод
1	1	Еспресо
2	1	Хот-дог
3	5	Капучіно
4	4	Латте
5	3	Американо
6	1	Мокко
7	3	Флет Вайт
8	2	Капучіно До
9	1	Шокочино

Рисунок Б.11 – Вікно «Продукція»

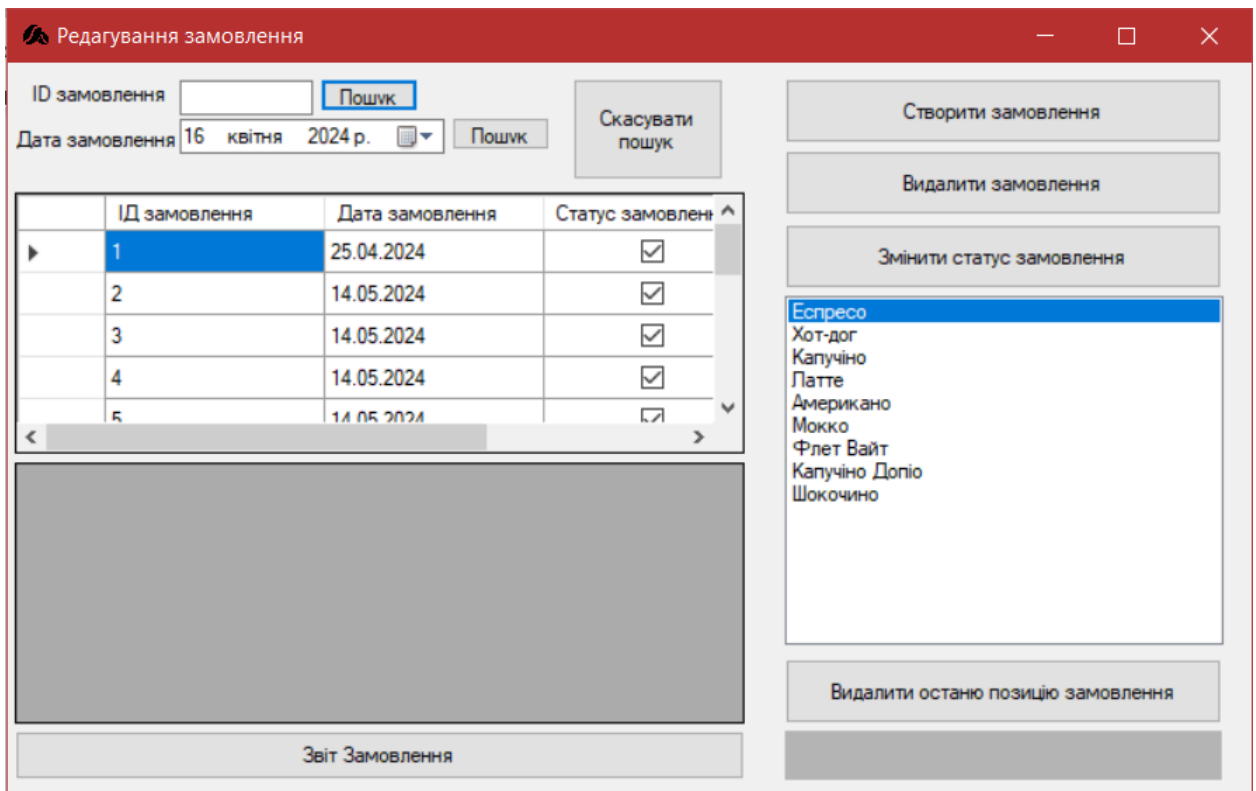


Рисунок Б.12 – Вікно «Редагування замовлення»



Рисунок Б.13 – Вікно «Графіки»

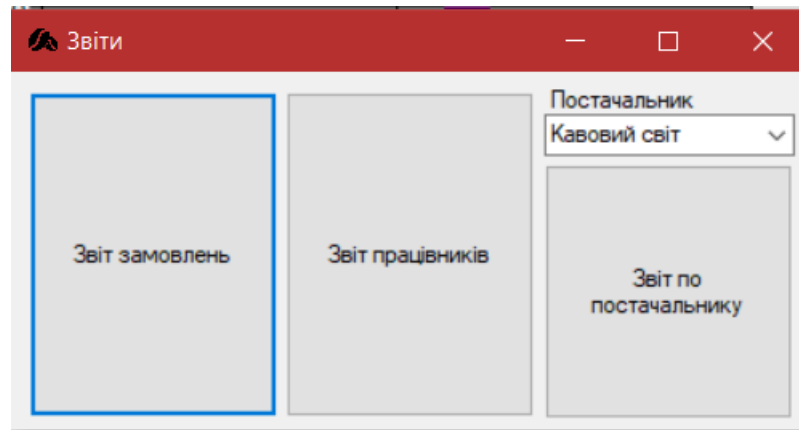


Рисунок Б.14 – Вікно «Звіти»

Звіт замовлень

ІД замовлення	Дата замовлення	Статус замовлення	ІД працівника
1	25.04.2024	<input checked="" type="checkbox"/>	2
2	14.05.2024	<input checked="" type="checkbox"/>	1
3	14.05.2024	<input checked="" type="checkbox"/>	1
4	14.05.2024	<input checked="" type="checkbox"/>	1
5	14.05.2024	<input checked="" type="checkbox"/>	1
6	19.05.2024	<input checked="" type="checkbox"/>	1
7	19.05.2024	<input type="checkbox"/>	1
8	19.05.2024	<input type="checkbox"/>	2
9	19.05.2024	<input checked="" type="checkbox"/>	2
10	20.05.2024	<input type="checkbox"/>	2
11	20.05.2024	<input type="checkbox"/>	1
12	23.05.2024	<input type="checkbox"/>	2
13	24.05.2024	<input type="checkbox"/>	1
14	26.05.2024	<input type="checkbox"/>	1

Рисунок Б.15 – Звіт «Звіт замовлень»

ООО «Кав'ярня»
Чек № 15
Касир: Петренко А.М.

1. Флет Вайт 1 95
2. Мокко 1 85
3. Капучіно Допіо 1 100

Каса: 1	
НДС	20%
Дата	2024-05-26 23:00:38
Підсумок	= 280 грн

Дякую за покупку!

Рисунок Б.16 – Звіт «Чек»

Додаток В. Код програми

```

using DGVPrinterHelper;
using Microsoft.Office.Interop.Word;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Application = Microsoft.Office.Interop.Word.Application;
using System.Xml.Linq;
using System.Data.SqlClient;
using static System.Windows.Forms.VisualStyleElement;
using DataTable = System.Data.DataTable;

namespace CafeProgram
{
    public partial class Filesmenu : Form
    {
        {string connectionString = "Data Source=DESKTOP-
Q4GQHJN\\SQLEXPRESS;Initial Catalog=Cafe;Integrated Security=True";
        public Filesmenu()
        {
            InitializeComponent();
            LoadSuppliers();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            var dgvPrinter = new DGVPrinter();
            dgvPrinter.CreateReport("Звіт замовлень", orderDataGridView);
        }
        private void orderBindingNavigatorSaveItem_Click(object sender,
        EventArgs e)
        {
            this.Validate();
            this.orderBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.cafeDataSet);
        }

        private void Filesmenu_Load(object sender, EventArgs e)
        {
            this.supplierTableAdapter.Fill(this.cafeDataSet.Supplier);
            this.employeesTableAdapter.Fill(this.cafeDataSet.Employees);
            this.orderTableAdapter.Fill(this.cafeDataSet.Order);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            var dgvPrinter = new DGVPrinter();
            dgvPrinter.CreateReport("Звіт працівників",
            employeesDataGridView);
        }
    }
}

```

```

private void LoadSuppliers()
{
    using (SqlConnection con = new SqlConnection(connectionString))
    {
        string query = "SELECT SupplierID, Supplier_Name FROM Supplier";
        SqlDataAdapter da = new SqlDataAdapter(query, con);
        DataTable dt = new DataTable();
        da.Fill(dt);

        comboBox1.DisplayMember = "Supplier_Name";
        comboBox1.ValueMember = "SupplierID";
        comboBox1.DataSource = dt; } }

private void comboBox1_SelectedIndexChanged(object sender, EventArgs
e)
{
    if (comboBox1.SelectedValue != null)
    {
        int selectedSupplierID = Convert.ToInt32(comboBox1.SelectedValue);
        LoadProductsBySupplier(selectedSupplierID); } }

private void LoadProductsBySupplier(int supplierID)
{
    using (SqlConnection con = new SqlConnection(connectionString))
    {
        string query = "SELECT ProductID, ProductName, Price,
Type_of_production FROM Production WHERE SupplierID = @SupplierID";
        SqlDataAdapter da = new SqlDataAdapter(query, con);
        da.SelectCommand.Parameters.AddWithValue("@SupplierID",
supplierID);
        DataTable dt = new DataTable();
        da.Fill(dt);

        dataGridView1.DataSource = dt;

        if (dataGridView1.Columns["ProductID"] != null)
        dataGridView1.Columns["ProductID"].HeaderText = "ІД продукції";
        if (dataGridView1.Columns["ProductName"] != null)
        dataGridView1.Columns["ProductName"].HeaderText = "Назва продукції";
        if (dataGridView1.Columns["Price"] != null)
        dataGridView1.Columns["Price"].HeaderText = "Ціна";
        if (dataGridView1.Columns["Type_of_production"] != null)
        dataGridView1.Columns["Type_of_production"].HeaderText = "Тип
продукції"; } }

private void button2_Click(object sender, EventArgs e)
{
    var dgvPrinter = new DGVPrinter();
    dgvPrinter.CreateReport("Звіт Постачальника", dataGridView1); } } }

```

```

public partial class Chart : Form
{
    string connectionString = "Data Source=DESKTOP-
Q4GQHJN\\SQLEXPRESS;Initial Catalog=Cafe;Integrated Security=True";
    public Chart()
    {InitializeComponent();}

    private void Chart_Load(object sender, EventArgs e)
    {
        chart1.Series.Clear();
        using (SqlConnection connection = new
SqlConnection(connectionString))
        {
            string query = "SELECT p.ProductName, SUM(od.Amount) AS
TotalAmount " + "FROM Production p " + "INNER JOIN [Order Details] od ON
p.ProductID = od.ProductID " + "GROUP BY p.ProductName";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            while (reader.Read())
            {
                string productName = reader["ProductName"].ToString();
                double totalAmount =
Convert.ToDouble(reader["TotalAmount"]);
                chart1.Series.Add(productName);
                chart1.Series[productName].Points.AddY(totalAmount);
            }
        }

        chart2.Series.Clear();

        using (SqlConnection connection = new
SqlConnection(connectionString))
        {
            string query = "SELECT MONTH(o.Date_of_order) AS Month,
SUM(od.Amount * p.Price) AS TotalProfit " + "FROM [Order] o " + "INNER
JOIN [Order Details] od ON o.OrderID = od.OrderID " + "INNER JOIN
Production p ON od.ProductID = p.ProductID " + "GROUP BY
MONTH(o.Date_of_order)";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            while (reader.Read())
            {
                int month = Convert.ToInt32(reader["Month"]);
                double totalProfit = Convert.ToDouble(reader["TotalProfit"]);
                string monthName = GetMonthName(month);
                chart2.Series.Add(monthName);
                chart2.Series[monthName].Points.AddY(totalProfit);
            }
        }

        chart3.Series.Clear();
    }
}

```

```

        using (SqlConnection connection = new
SqlConnection(connectionString))
        {string query = "SELECT e.FIO, COUNT(o.Employee_ID) AS
TotalOrders " + "FROM Employees e " + "LEFT JOIN [Order] o ON
e.Employee_ID = o.Employee_ID " + "GROUP BY e.FIO";
        SqlCommand command = new SqlCommand(query, connection);
        connection.Open();
        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {string employeeName = reader["FIO"].ToString();
        int totalOrders = Convert.ToInt32(reader["TotalOrders"]);
        chart3.Series.Add(employeeName);
        chart3.Series[employeeName].Points.AddY(totalOrders);} } }

private string GetMonthName(int monthNumber)
{DateTimeFormatInfo ukrainianCulture = new CultureInfo("uk-UA",
false).DateTimeFormat;
    string monthName = ukrainianCulture.GetMonthName(monthNumber);
    return monthName;}

```

В.2 Код форми «Графіки»

```

public partial class OrderFind : Form
{string connectionString = "Data Source=DESKTOP-
Q4GQHJN\\SQLEXPRESS;Initial Catalog=Cafe;Integrated Security=True";
    public OrderFind()
    {InitializeComponent();
    LoadAllOrders();
    LoadProductionListBox();}

private void OrderFind_Load(object sender, EventArgs e)
{this.order_DetailsTableAdapter.Fill(this.cafeDataSet.Order_Details);
    this.productionTableAdapter.Fill(this.cafeDataSet.Production);
    this.orderTableAdapter.Fill(this.cafeDataSet.Order);}

private void Searchbutton_Click(object sender, EventArgs e)
{ClearOrderDetails();
    int orderId;
    if (int.TryParse(IDsearchBox.Text, out orderId))
    {SearchByOrderID(orderId);}
    else
    {MessageBox.Show("Please enter a valid Order ID.");}}

private void SearchbuttonDate_Click(object sender, EventArgs e)
{ClearOrderDetails();
    DateTime selectedDate = dateTimePicker1.Value.Date;
}

```

```

        SearchByOrderDate(selectedDate);}

private void CancelSearch_Click(object sender, EventArgs e)
{ ClearOrderDetails();
  LoadAllOrders();}

private void SearchByOrderID(int orderId)
{ string query = "SELECT * FROM [Order] WHERE OrderID = @OrderID";
  using (SqlConnection conn = new SqlConnection(connectionString))
  { SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
    adapter.SelectCommand.Parameters.AddWithValue("@OrderID",
orderId);
    DataTable dataTable = new DataTable();
    adapter.Fill(dataTable);
    OrderGridView.DataSource = dataTable; } }

private void SearchByOrderDate(DateTime orderDate)
{ string query = "SELECT * FROM [Order] WHERE Date_of_order =
@Date_of_order";
  using (SqlConnection conn = new SqlConnection(connectionString))
  { SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
adapter.SelectCommand.Parameters.AddWithValue("@Date_of_order",
orderDate);
    DataTable dataTable = new DataTable();
    adapter.Fill(dataTable);
    OrderGridView.DataSource = dataTable; } }

private void LoadAllOrders()
{ string query = "SELECT * FROM [Order]";
  using (SqlConnection conn = new SqlConnection(connectionString))
  { SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
    DataTable dataTable = new DataTable();
    adapter.Fill(dataTable);
    OrderGridView.DataSource = dataTable; } }

private void LoadOrderDetails(int orderId)
{ { string query = @"
SELECT od.SubOrderID AS 'ІД позиції',
      od.OrderID,
      p.ProductName AS 'Продукт',
      od.Amount AS 'Кількість'
FROM [Order Details] od
JOIN Production p ON od.ProductID = p.ProductID
WHERE od.OrderID = @OrderID";

```

```

using (SqlConnection conn = new SqlConnection(connectionString))
{SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
  adapter.SelectCommand.Parameters.AddWithValue("@OrderID",
orderId);
  DataTable dataTable = new DataTable();
  adapter.Fill(dataTable);
  OrderDetailsGridView.DataSource = dataTable;
  if (OrderDetailsGridView.Columns["OrderID"] != null)
  { OrderDetailsGridView.Columns["OrderID"].Visible = false; } } }

private void OrderGridView_SelectionChanged(object sender, EventArgs e)
{ try
  { if (OrderGridView.SelectedRows.Count > 0)
    { int orderId
=Convert.ToInt32(OrderGridView.SelectedRows[0].Cells["orderIDDataGridViewTextBoxColumn"].Value);
      LoadOrderDetails(orderId); } } catch { } }
private void ClearOrderDetails()
{ OrderDetailsGridView.DataSource = null; }

private void btnCreateOrder_Click(object sender, EventArgs e)
{ using (SqlConnection conn = new SqlConnection(connectionString))
  { conn.Open();
    string query = "SELECT ISNULL(MAX(OrderID), 0) + 1 FROM
[Order]";
    SqlCommand cmd = new SqlCommand(query, conn);
    int newOrderId = (int)cmd.ExecuteScalar();
    query = @"INSERT INTO [Order] (OrderID, Date_of_order,
Status_of_order, Employee_ID) VALUES (@OrderID, @Date_of_order,
@Status_of_order, @Employee_ID)";
    cmd = new SqlCommand(query, conn);
    cmd.Parameters.AddWithValue("@OrderID", newOrderId);
    cmd.Parameters.AddWithValue("@Date_of_order",
DateTime.Now.ToString("yyyy-MM-dd"));
    cmd.Parameters.AddWithValue("@Status_of_order", false);
    cmd.Parameters.AddWithValue("@Employee_ID",
AppState.PositionID);

    cmd.ExecuteNonQuery(); }
  LoadAllOrders(); }

private void btnDeleteOrder_Click(object sender, EventArgs e)
{ if (OrderGridView.SelectedRows.Count > 0)

```

```

        {int                orderId                =
Convert.ToInt32(OrderGridView.SelectedRows[0].Cells["orderIDDataGridView
TextBoxColumn"].Value);
        using (SqlConnection conn = new SqlConnection(connectionString))
        {conn.Open();
            string query = "DELETE FROM [Order Details] WHERE OrderID =
@OrderID";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@OrderID", orderId);
            cmd.ExecuteNonQuery();
            query = "DELETE FROM [Order] WHERE OrderID = @OrderID";
            cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@OrderID", orderId);
            cmd.ExecuteNonQuery();
        }
        LoadAllOrders();
        ClearOrderDetails();} }

```

```

private void btnChangeOstatus_Click(object sender, EventArgs e)
{ if (OrderGridView.SelectedRows.Count > 0)
    {int                orderId                =
Convert.ToInt32(OrderGridView.SelectedRows[0].Cells["orderIDDataGridView
TextBoxColumn"].Value);
        bool                currentStatus                =
Convert.ToBoolean(OrderGridView.SelectedRows[0].Cells["statusoforderDataG
ridViewCheckBoxColumn"].Value);
        bool newStatus = !currentStatus;
        using (SqlConnection conn = new SqlConnection(connectionString))
        {conn.Open();
            string query = "UPDATE [Order] SET Status_of_order =
@Status_of_order WHERE OrderID = @OrderID";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@Status_of_order", newStatus);
            cmd.Parameters.AddWithValue("@OrderID", orderId);
            cmd.ExecuteNonQuery();
        }
        LoadAllOrders();} }

```

```

private void LoadProductionListBox()
{ string query = "SELECT ProductID, ProductName FROM Production";
    using (SqlConnection conn = new SqlConnection(connectionString))
    {SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
        DataTable dataTable = new DataTable();
        adapter.Fill(dataTable);
        ProductionListBox.DisplayMember = "ProductName";
        ProductionListBox.ValueMember = "ProductID";
        ProductionListBox.DataSource = dataTable;} }

```

```

private void btnDeletelastSubOrder_Click(object sender, EventArgs e)
{
    if (OrderGridView.SelectedRows.Count > 0)
    {
        int orderId = Convert.ToInt32(OrderGridView.SelectedRows[0].Cells["orderIDDataGridView
        TextBoxColumn"].Value);
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            conn.Open();
            string query = "SELECT TOP 1 SubOrderID FROM [Order Details]
            WHERE OrderID = @OrderID ORDER BY SubOrderID DESC";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@OrderID", orderId);
            object result = cmd.ExecuteScalar();
            if (result != null)
            {
                int lastSubOrderId = (int)result;
                query = "DELETE FROM [Order Details] WHERE SubOrderID =
                @SubOrderID";
                cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@SubOrderID",
                lastSubOrderId);
                cmd.ExecuteNonQuery();
            }
            LoadOrderDetails(orderId);
        }
    }
}

private void ProductionListBox_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (OrderGridView.SelectedRows.Count > 0)
    {
        int orderId = Convert.ToInt32(OrderGridView.SelectedRows[0].Cells["orderIDDataGridView
        TextBoxColumn"].Value);
        int productId = Convert.ToInt32(ProductionListBox.SelectedValue);
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            conn.Open();
            string query = "SELECT COUNT(*) FROM [Order Details] WHERE
            OrderID = @OrderID AND ProductID = @ProductID";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@OrderID", orderId);
            cmd.Parameters.AddWithValue("@ProductID", productId);
            int count = (int)cmd.ExecuteScalar();

            if (count > 0)
            {
                query = "UPDATE [Order Details] SET Amount = Amount + 1
                WHERE OrderID = @OrderID AND ProductID = @ProductID";
                cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@OrderID", orderId);
                cmd.Parameters.AddWithValue("@ProductID", productId);
            }
        }
    }
}

```

```

        cmd.ExecuteNonQuery();
    else
        {query = @"INSERT INTO [Order Details] (SubOrderID, OrderID,
ProductID, Amount)
        VALUES ((SELECT ISNULL(MAX(SubOrderID), 0) + 1
FROM [Order Details] WHERE OrderID = @OrderID), @OrderID, @ProductID,
1)";
        cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@OrderID", orderId);
        cmd.Parameters.AddWithValue("@ProductID", productId);
        cmd.ExecuteNonQuery();}
    LoadOrderDetails(orderId);}

private void OrderReport_Click(object sender, EventArgs e)
{if (OrderGridView.SelectedRows.Count > 0)
    {int                orderId                =
Convert.ToInt32(OrderGridView.SelectedRows[0].Cells["orderIDDataGridView
TextBoxColumn"].Value);
    string reportTitle = "№" + orderId + " замовлення";
    var dgvPrinter = new DGVPrinter();
    dgvPrinter.CreateReport(reportTitle, OrderDetailsGridView);}
else
    {MessageBox.Show("Будь ласка, спочатку виберіть замовлення для
створення звіту.");}}}}

```

В.3 Код форми «Редагування замовлення»

```

namespace CafeProgram
{public partial class Authorization : Form
    {public Authorization()
        {InitializeComponent();
        int positionID = AppState.PositionID;}

    private void button1_Click(object sender, EventArgs e)
    {textBox2.UseSystemPasswordChar =
!textBox2.UseSystemPasswordChar;}

    private void AuthorizationForm_Shown(object sender, EventArgs e)
    {textBox2.UseSystemPasswordChar = true;}

    private void Enterbutton_Click(object sender, EventArgs e)
    {string username = textBox1.Text;
    string password = textBox2.Text;
    string connectionString = "Data Source=DESKTOP-
Q4GQHJN\\SQLEXPRESS;Initial Catalog=Cafe;Integrated Security=True";
    using (SqlConnection conn = new SqlConnection(connectionString))

```

```

{conn.Open();

    string query = "SELECT ID_position, FIO FROM Employees
WHERE username = @username AND password = @password";
    using (SqlCommand cmd = new SqlCommand(query, conn))
    {cmd.Parameters.AddWithValue("@username", username);
      cmd.Parameters.AddWithValue("@password", password);

      using (SqlDataReader reader = cmd.ExecuteReader())
      {if (reader.Read()) // якщо дані були знайдені
        {int positionID = reader.GetInt32(0); // перший стовпець
(ID_position)
          string fio = reader.GetString(1); // другий стовпець (FIO)
          AppState.PositionID = positionID;
          AppState.FIO = fio;

          var menuForm = new Menu();
          menuForm.Show();
          this.Hide();}
        else
        {MessageBox.Show("Логін чи пароль були введені
невірно.");}}}}}}

    private void button2_Click(object sender, EventArgs e)
    {var RecoverForm = new LOPRecovery();
      RecoverForm.Show();}

    private void Authorization_FormClosing(object sender,
FormClosingEventArgs e)
    {Application.Exit();}}

public static class AppState
{public static int PositionID { get; set; }
  public static string FIO { get; set; }}}

```

В.4 Код форми «Авторизація»

```

namespace CafeProgram
{
    public partial class Schedule : Form
    {
        private string connectionString = "Data Source=DESKTOP-
        Q4GQHJN\\SQLEXPRESS;Initial Catalog=Cafe;Integrated Security=True";

        private Dictionary<string, string> UkrainianDays = new Dictionary<string,
        string>()
        {
            {"Monday", "Понеділок"}, {"Tuesday", "Вівторок"}, {"Wednesday",
            "Середа"}, {"Thursday", "Четвер"}, {"Friday", "П'ятниця"}, {"Saturday",
            "Субота"}, {"Sunday", "Неділя"};
        }

        public Schedule()
        {
            InitializeComponent();
            this.Load += new EventHandler(Schedule_Load);
        }

        private void Schedule_Load(object sender, EventArgs e)
        {
            this.scheduleTableAdapter.Fill(this.cafeDataSet.Schedule);
            CopyEmployeesToSchedule();
            int positionID = (int)AppState.PositionID;
            if (positionID == 2)
            {
                dataGridView1.Enabled = false;
                button1.Enabled = false;
                button2.Enabled = false;
            }
            else
            {
                dataGridView1.Enabled = true;
                button1.Enabled = true;
                button2.Enabled = true;
            }
            TranslateColumnHeaders();
        }

        private void TranslateColumnHeaders()
        {
            foreach (DataGridViewColumn column in dataGridView1.Columns)
            {
                if (UkrainianDays.ContainsKey(column.HeaderText))
                {
                    column.HeaderText = UkrainianDays[column.HeaderText];
                }
            }
        }

        private void CopyEmployeesToSchedule()
        {
            string query = @"
            INSERT INTO Schedule (Employee_ID, FIO)
            SELECT Employee_ID, FIO
            FROM Employees
            WHERE Employee_ID NOT IN (SELECT Employee_ID FROM
            Schedule)";

            using (SqlConnection connection = new
            SqlConnection(connectionString))
            {
                connection.Open();
            }
        }
    }
}

```

```

        SqlCommand command = new SqlCommand(query, connection);
        int rowsAffected = command.ExecuteNonQuery();} }

private void button1_Click(object sender, EventArgs e)
{this.Validate();
  this.scheduleBindingSource.EndEdit();
  this.scheduleTableAdapter.Update(this.cafeDataSet);}

private void button2_Click(object sender, EventArgs e)
{ClearWeekDays();}

private void ClearWeekDays(object employeeID = null)
{string query = @"
  UPDATE Schedule
  SET Monday = NULL,
    Tuesday = NULL,
    Wednesday = NULL,
    Thursday = NULL,
    Friday = NULL,
    Saturday = NULL,
    Sunday = NULL";

  using (SqlConnection connection = new
SqlConnection(connectionString))
  {connection.Open();
    SqlCommand command = new SqlCommand(query, connection);
    command.ExecuteNonQuery();}
  this.Validate();
  this.scheduleBindingSource.EndEdit();
  this.scheduleTableAdapter.Update(this.cafeDataSet);
  this.scheduleTableAdapter.Fill(this.cafeDataSet.Schedule);
  this.scheduleTableAdapter.Update(this.cafeDataSet);}

private void button3_Click(object sender, EventArgs e)
{var dgvPrinter = new DGVPrinter();
  dgvPrinter.CreateReport("Розклад", dataGridView1);}} }

```

В.5 Код форми «Розклад»

Додаток Г. Концепт програми

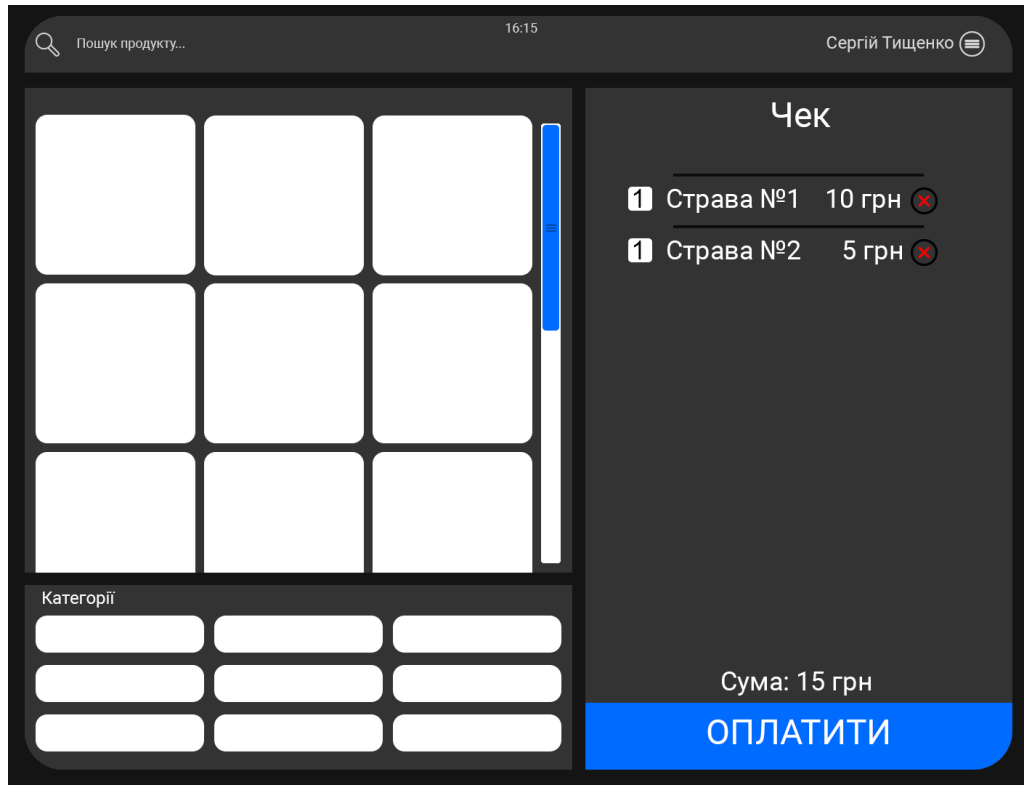


Рисунок Г.1. — концепт форми каси

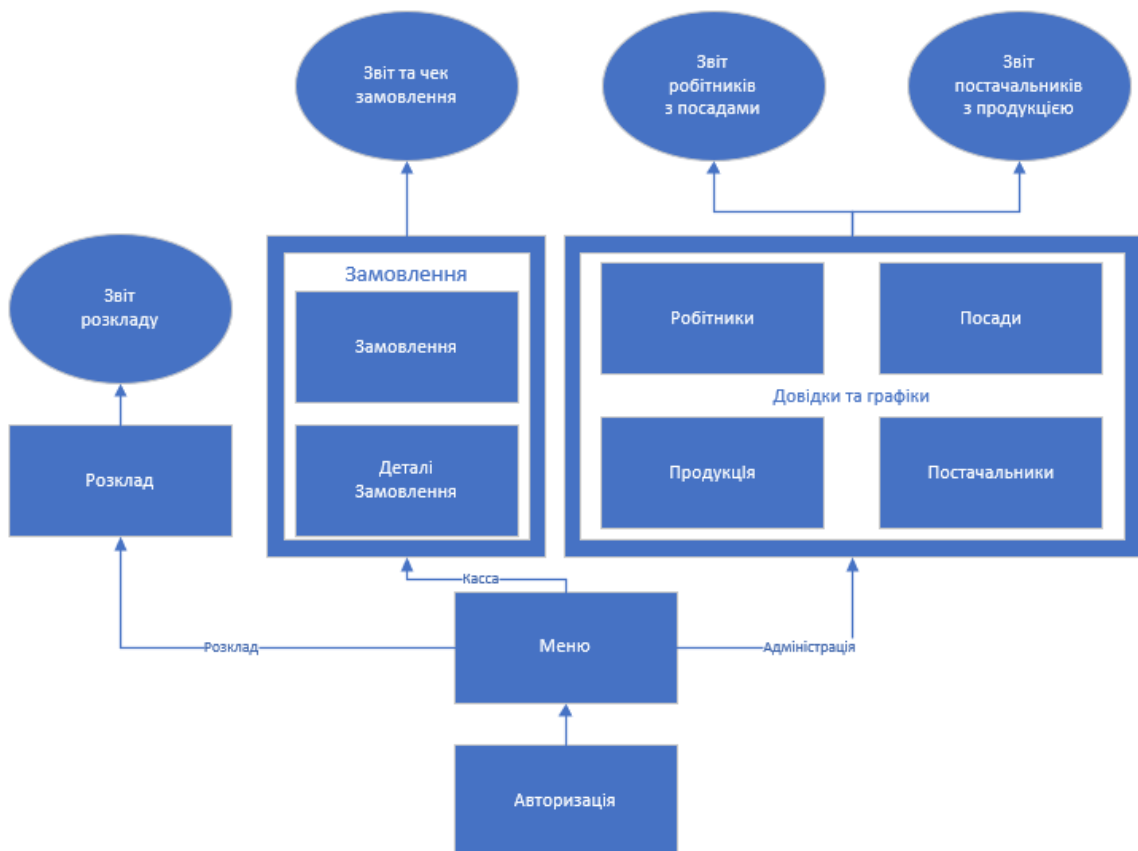


Рисунок Г.2. — концепт схема програми