

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Яценко Віталія Сергійовича
(ПІБ)

академічної групи 122-19-1
(шифр)

спеціальності 122 Комп'ютерні науки
(код і назва спеціальності)

освітньої програми Комп'ютерні науки
(назва освітньої програми)

на тему: Розробка автоматизованої системи аналізу ефективності торговельного підприємства

| Керівники | Прізвище, ініціали | Оцінка за шкалою | | Підпис |
|------------------------|----------------------------|------------------|---------------|--------|
| | | рейтинговою | інституційною | |
| кваліфікаційної роботи | | | | |
| розділів: | | | | |
| спеціальний | <i>Проф. Мороз Б.І.</i> | | | |
| економічний | <i>Проф. Вагонова О.Г.</i> | | | |
| | | | | |
| | | | | |
| Рецензент | | | | |
| Нормоконтролер | <i>Доц. Гуліна І.Г.</i> | | | |

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

« »

2023 року

ЗАВДАННЯ

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-19-1 Яценко В.С

(група)

(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка Автоматизованої системи аналізу

ефективності торгівельного підприємства

затверджена наказом ректора НТУ «ДП» від

16.05.2023

№

| Розділ | Зміст виконання | Термін виконання |
|-------------|--|-------------------|
| Спеціальний | <i>На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i> | 13.05.2023 |
| Економічний | <i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПО й тривалості його розробки</i> | 27.05.2023 |

Завдання видав

(підпис)

Проф. Мороз Б.І

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Яценко В.С

(прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

РЕФЕРАТ

Пояснювальна записка: с.79, рис.8 , дод.3 , джерел 20.

Об'єкт розробки: додаток для аналізу комерційної ефективності підприємства.

Мета кваліфікаційної роботи: надати кінцевому користувачу інструмент для аналізу ефективності підприємства за певними показниками для прийняття рішень, які допоможуть підняти прибуток.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної області, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування підсистеми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження застосунку, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню застосунку та розраховано час на його створення.

Практичне значення полягає в наданні користувачу інструменту для аналізу комерційної ефективності.

Актуальність даної роботи полягає в аналізі даних, які здобуваються з фінансових звітів. Завдяки аналізу даних кінцевий користувач, який є власником торговельного підприємства, зможе виявити фактори, які негативно впливають на прибуток, та прийняти рішення для його збільшення. Актуальність даного типу додатків визначається великим попитом на подібні розробки.

Список ключових слів: Аналітична система, ефективність торгової організації, алгоритм аналізу і досліджень, моделювання і прогнозування результатів.

ABSTRACT

Explanatory note: pp.79, fig.8 , appendices 3, sources 20.

Object of development: application for analyzing of efficiency of commercial company.

The purpose of the qualification work: to provide the user with a tool for commercial analysis through the development and use of appropriate application.

The introduction considers the analysis and current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development are defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes the existing solutions, selects the platform for development, performs design and development of the program, describes the algorithm and structure of the subsystem, determines the input and output data, provides characteristics of the parameters of technical means, describes the call and application load, describes the program.

The economic section determines the complexity of the developed information system, calculates the cost of work to create an application and calculates the time to create it.

Of practical importance is to provide the user with a tool to analyze how effective the enterprise.

The relevance of this work is that analyzing data, which is gained from financial report. Thanks to that data the end user, who is the actual company owner, can find the factors, which affects negatively on income and make a decision, which can increase it. Relevance of that kind app is determined by the big require for these developments.

Keywords: Analytical system, effectiveness of business operations, algorithm analysis and research, modeling and forecasting results.

ЗМІСТ

| | |
|---|--|
| РЕФЕРАТ..... | |
| ABSTRACT..... | |
| СПИСОК УМОВНИХ ПОЗНАЧЕНЬ..... | |
| ВСТУП..... | |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ..... | |
| 1.1 Загальні відомості з предметної галузі..... | |
| 1.2 Призначення розробки та область застосування..... | |
| 1.3 Підстава для розробки..... | |
| 1.4 Постановка завдання..... | |
| 1.5 Вимоги до програми або програмного виробу..... | |
| 1.5.1 Вимоги до функціональних характеристик | |
| 1.5.2 Вимоги до інформаційної безпеки..... | |
| 1.5.3 Вимоги до складу та параметрів технічних засобів..... | |
| 1.5.4 Вимоги до інформаційної та програмної сумісності..... | |
| РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ..... | |
| 2.1 Функціональне призначення системи..... | |
| 2.2 Опис застосованих математичних методів..... | |
| 2.3 Опис використаних технологій та мов програмування..... | |
| 2.4 Опис структури системи та алгоритмів її функціонування..... | |
| 2.5 Обґрунтування та організація вхідних та вихідних даних програми..... | |
| 2.6 Опис роботи розробленої системи..... | |
| 2.6.1 Використані технічні засоби..... | |
| 2.6.2 Використані програмні засоби..... | |
| 2.6.3 Виклик та завантаження програми..... | |

| | |
|---|---|
| 2.6.4 | Опис інтерфейсу користувача..... |
| РОЗДІЛ 3. ЕКОНОМІКА..... | |
| 3.1 | Розрахунок трудомісткості та вартості розробки програмного продукту |
| 3.2 | Розрахунок витрат на створення програми..... |
| ВИСНОВКИ..... | |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | |
| Додаток А. Код програми..... | |
| Додаток Б. Відзив керівника економічного розділу..... | |
| Додаток В. Перелік файлів на диску..... | |

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

- БД - база даних
- ЕОМ - електронно-обчислювальна машина
- ІС - інформаційна система
- ПЗ - Програмне забезпечення
- СУБД - система управління базами даних
- ІТ - Інформаційні технології
- SEO - пошукова оптимізація

ВСТУП

Будь-яке підприємство засновується для отримання прибутку, яке потім розподіляється на оплату аренди, зарплату працівникам, податки, страхування, постачання товару та інші витрати, на які підприємство виділяє кошти. Але потрібно звертати увагу на те, який прибуток приносить те саме підприємство, чи достатньо коштів йому вийти хоча б в нуль, аби не закритися та розробити новий фінансовий план, щоб збільшити прибуток.

В основному ефективність підприємства оцінюють за 6 факторами:

- Конверсія
- Середній чек
- Продажи з квадратного метру торгової площі
- Об'єм продажу в валюті й натуральних показниках
- Повернення
- Зарплатомісткість

Завдяки цим факторам можна визначити, наскільки ефективно працює підприємство й дає можливість зрозуміти, чи не потрібні зміни для збільшення прибутку, або взагалі воно не несе ніякої комерційної користі.

У кваліфікаційній роботі розробляється:

- Клієнтський додаток – аналізатор комерційної ефективності.
- База даних користувачів, які мають доступ до підприємства.

Актуальність даної роботи полягає в тому, що контроль ефективності підприємства завжди має найвищий пріоритет для власника, який буде користуватися додатком та визначати фактори, які впливають на прибуток й робити зміни, які допоможуть його підняти.

Мета роботи – розробити додаток для фінансового аналізу. Ця програма повинна мати інтуїтивний інтерфейс користувача, надавати усі необхідні функції та показники прибутку та витрат.

З поставленої мети впливають завдання:

- створити структуру бази даних;
- створити архітектуру проекту;
- вивчити програми-конкуренти, аналоги;
- Провести аналіз за допомогою аналізуючої програм та провести пробну статистику.

статистику.

- створити зручний та зрозумілий інтерфейс користувача.

Також дослідження затребуваності додатка фінансового аналізу, ознайомлення з тенденціями аналізу, вивчення способів інтерактивної взаємодії з користувачем у додатку.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Аналізуючи функціонал та інтерфейс сайтів-аналогів, можна дійти висновку, що більшість подібних пропозицій орієнтована на спортсменів, чие харчування та спосіб життя певною мірою відрізняється від способу життя та харчування звичайного користувача.

Крім цього, важливо врахувати неадаптованість під різні країни та регіони. Найчастіше досить складно адаптувати систему під економіку країни користувача, яка постійно змінюється.

Додатки – аналоги:

- Deductor. Аналітична платформа для ефективного рішення проблеми тиражування знань. Не дивлячись на старий інтерфейс, є одним з найефективніших платформ для аналізу даних. Технології даної платформи дозволяють проходити усі етапи побудування аналітичної системи: від створення сховища даних до автоматичного підбору моделей й візуалізації даних. Особливості:

- Вбудований штучний інтелект
 - Весь інструментарій вбудований в одну платформу
 - Вбудована інтеграція з десятками джерел даних
 - Висока продуктивність
 - Аналітика від простих формул до алгоритмів, які навчаються самі.
 - Розробка сценаріїв аналізу без сторонньої допомоги технічних спеціалістів.
- SPSS (Statistical Package for the Social Sciences) Statistics. Програма, яка дозволяє проводити розвинутий статистичний аналіз даних, а також має функцію від планування й збору даних до аналізу й побудови бізнес – звіту. Має інтуїтивний та приємний для очей інтерфейс, систему візуалізації через різні графіки. Але є певні недоліки:
 - Не може використовуватися з публічної хмари (SaaS)
 - Відсутня безкоштовна версія. Тільки пробна демонстраційна на певний час.
 - Відсутній штучний інтелект для спрощення процесу обробки.
 - AmazonGO. Не зважаючи на те, що це представляє собою додаток, який призначений для допомоги покупцю в покупці товарів, він дає вхідні дані стосовно відвідувача та його кошику, що він набирає за час відвідування магазину. Система працює таким чином: відвідувач на вході в магазин вноситься в базу даних за допомогою сенсорів та камер. Потім дана система слідує за ним та аналізує всі товари, які він бере у свій кошик, та робить список придбань онлайн. Після того, як відвідувач закінчив обхід магазину, приходячи до каси, система автоматично виписує йому чек та кошти з його рахунку.^[1]

1.2. Призначення розробки та область застосування

Розробка програмного додатку призначено для аналізу комерційної ефективності підприємства.

Аналізуючи дані про продажі груп товарів, можна зробити висновок, в який період певна група приносить більше прибутку. Завдяки цьому можна покращити якість обслуговування та товарів, які продаються. Дана програма через графіки показує скільки було продано товару за певний період.

1.3. Підстава для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу. Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № [] від [].2023 р;
- завдання на кваліфікаційну роботу на тему: «Розробка автоматизованої системи аналізу ефективності торгівельного підприємства».

1.4. Постановка завдання

Мета роботи – розробити додаток, що буде брати дані з бази даних, в яку будуть вноситися покупки товарів, їх кількість, група товару. Через ці дані програма буде аналізувати наскільки певний період був прибутковий, скільки було отримано через продажі та скільки було витрачено на оплату підприємства, аренду приміщення, оплату персоналу, оплату постачання товарів, податку для держави.

З поставленої задачі стоїть:

- створити базу даних;
- продумати схему аналізу, через яку програма буде обробляти дані;
- Вивчити, дослідити програми – конкуренти, аналоги;
- Створити інтерфейс з графіками статистики.;

Також дослідити запит на ринку у затребуваності додатків комерційного аналізу.

Впровадження функціоналу дослідження підприємства та його комерційної ефективності.

Основні розділи програми повинні бути наступними:

- Вікно введення логіну та паролю;
- Головна сторінка с графіками про продажі;
- Відділ сумарних показників за період;
- Відділ витрат на підприємство;
- Відділ товарів;
- База даних користувачів (для режиму адміністратора).

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Система повинна мати зручний та інтуїтивно зрозумілий інтерфейс та забезпечувати наведені нижче функціональні характеристики.

Так як додаток буде використовуватися власником підприємства, доступом до даних мають володіти він сам, його економісти й бухгалтера. Саме тому потрібно зробити систему авторизації, яка буде їх ідентифікувати та давати їм доступ.

Авторизація являє собою вхід з використанням логіну та паролю.

До авторизації користувач не може взаємодіяти з програмою, відкривається тільки вікно авторизації.

При цьому потрібно зауважити: при першому запуску програма запросить логін та пароль для створення логіну з правами адміністратора, який потім зможе керувати базою даних, створюючи або видаляючи користувачів. Після авторизації відкривається доступ до всіх розділів.

У розділі «Витрати» будуть показуватися усі витрати на підприємство: оплата персоналу, оплата аренди, оплата податку з продажів та інші витрати.

У розділі «Товари» будуть вводитися ціни продажу та закупівлі для постачання для збуту до кінцевого споживача. В даному розділі користувач бачить ціну постачання та націнку продажу у магазині, за яку покупці платять.

У розділі «Додати користувача» користувач з правами адміністратора контролює усіх інших користувачів, які користуються програмою, та розпоряджається ними: додає або видаляє користувачів з бази даних.

У розділі «Фінансовий звіт» розкривається інформація, наскільки підприємство ефективно працює за певний період, скільки було витрачено грошей на саме підприємство, який був чистий прибуток, площа приміщення підприємства та прибуток на кожен метр квадратний цього підприємства.

Таким чином визначається, чи добре виконує своє завдання підприємство чи його вигідніше закрити.

1.5.2 Вимоги до інформаційної безпеки

Для уникнення некоректної роботи програми необхідно реалізувати:

- семантичний та синтаксичний контроль вхідних даних;
- обробку виняткових ситуацій;
- виведення повідомлень про помилки;
- можливість повторного введення даних;
- можливість безперервної роботи протягом не менше 120 годин (5 діб);

Як складовий елемент системи, пов'язаної із торгівельною діяльністю, програма повинна мати:

- час відновлення після збою - 5 хв;
- час відновлення після відмови одного з елементів підсистеми не повинно перевищувати половини операційного банківського дня;
- вірогідність виникнення не більше 2 логічних помилок на 1000 операторів за 1 рік експлуатації;
- забезпечення неушкодженого стану даних, що зберігаються в базі даних, у випадку відмови підсистеми.

1.5.3 Вимоги до складу та параметрів технічних засобів

Для нормального функціонування програми необхідно, щоб обчислювальна машина, на якій буде функціонувати клієнтській додаток, відповідала наступним вимогам:

- процесор AMD Ryzen 5 3600 3,59 GHz;
- не менше 16GB оперативної пам'яті;
- рідкокристалічний монітор з діагоналлю не менше 16”;
- 100 мб вільного місця на жорсткому диску;

- клавіатура;
- маніпулятор «миша»;

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених, розроблений програмний виріб буде функціонувати відповідно до вимог щодо надійності, швидкості обробки даних і безпеки, висунутими замовником.

З точки зору користувачів програма не є вимогливою до складу та параметрів технічних засобів та може завантажуватись на ПК та ноутбуках під управлінням ОС Windows.

1.5.4 Вимоги до інформаційної та програмної сумісності

Програма не є вимогливою до складу та параметрів технічних засобів та може завантажуватись на ПК, ноутбуках, на ОС Windows.

Вимоги до програмних характеристик сервера:

- Підтримка SQLite

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Система повинна мати зручний та інтуїтивно зрозумілий інтерфейс та забезпечувати наведені нижче функціональні характеристики.

Так як додаток буде використовуватися власником підприємства, доступом до даних мають володіти він сам, його економісти й бухгалтера. Саме тому потрібно зробити систему авторизації, яка буде їх ідентифікувати та давати їм доступ.

Авторизація являє собою вхід з використанням логіну та паролю.

До авторизації користувач не може взаємодіяти з програмою, відкривається тільки вікно авторизації.

При цьому потрібно зауважити: при першому запуску програма запросить логін та пароль для створення логіну з правами адміністратора, який потім зможе керувати базою даних, створюючи або видаляючи користувачів. Після авторизації відкривається доступ до всіх розділів.

На головній сторінці знаходиться таблиця БД, 3 графіка, які відображають оброблені дані, кнопки для налаштування графіку, що будуть вказувати, що можна показати на графіці та розділи, які будуть описані нижче.

У розділі «Витрати» будуть показуватися усі витрати на підприємство: оплата персоналу, оплата аренди, оплата податку з продажів та інші витрати.

У розділі «Товари» будуть вводитися ціни продажу та закупівлі для постачання для збуту до кінцевого споживача. В даному розділі користувач бачить ціну постачання та націнку продажу у магазині, за яку покупці платять.

У розділі «Додати користувача» користувач з правами адміністратора контролює усіх інших користувачів, які користуються програмою, та розпоряджається ними: додає або видаляє користувачів з бази даних.

У розділі «Фінансовий звіт» розкривається інформація, наскільки підприємство ефективно працює за певний період, скільки було витрачено грошей на саме підприємство, який був чистий прибуток, площа приміщення підприємства та прибуток на кожен метр квадратний цього підприємства. Таким чином визначається, чи добре виконує своє завдання підприємство чи його вигідніше закрити.

2.2. Опис застосованих математичних методів

Розрахунок конверсії. Конверсія – це співвідношення людей, які відвідували магазин та зробили реальні покупки в день. Це є показником професіоналізму продавців – консультантів. Чим вище цей показник, тим краще працює торговий персонал.

Приклад: в магазин зайшло 178 людей. З них кількість придбань (чеків) – 51. Щоб визначити цей показник, потрібно кількість чеків поділити на загальну кількість людей й помножити на 100%.

$$K = \frac{\text{Кількість чеків } (N)}{\text{Загальна кількість людей } (L)} * 100\% = \frac{51}{178} * 100\% = 28,6\%$$

Слід зазначити, що зміни для даного показнику – є нормою. Але тільки коли він показує позитивні результати, які проявляються в підвищенні. Якщо конверсія не знижується або взагалі не змінюється – потрібно провести аналіз можливих причин цьому.

Розрахунок прибутку на м². Маючи певне приміщення, необхідно розраховувати прибуток на кожний його метр. Розрахунок даного показника бере дані з виручки за певний період (день, тиждень, місяць) ділений на площу торгового приміщення в квадратних метрах (м²).

Приклад: магазин має приміщення площею 150 кв.метрів. За місяць він отримав 362 700 гривень. Поділивши прибуток на площу, отримаємо 2 418 на м².

$$P_{M^2} = \frac{\text{Виручка за певний період (день,тиждень,місяць)}}{\text{Площа торгового приміщення}} = \frac{362\,700}{150} = 2\,418 \text{ грн/м}^2$$

2.3. Опис використаних технологій та мов програмування

При розробці ПЗ була використана мова програмування C# - (читається як «сі шарп») – проста, потужна, статично типізована, об’єктно орієнтована мова програмування від компанії Microsoft. C# входить до сімейства мов програмування C, синтаксис мови буде знайомим програмістам, що працювали з C, C++, Java та JavaScript.

Перша версія мови C# була створена в 1998 – 2001 роках, групою інженерів Microsoft під керівництвом Андреса Гейлсберга та Скотта Вільтамота, як основна мова програмування платформи Microsoft .Net.

C# увібрав в себе найкращі властивості попередників – мов C, C++, Modula, Object Pascal, спираючись на практичний досвід їх використання. Деякі проблематичні моделі, що до цього використовувались у мовах програмування, зокрема множинне спадкування класів (яке використовується у мові C++), були свідомо виключені.

У багатьох відношеннях мова C# дуже схожа на Java, це відображено в синтаксисах та основних поняттях цих мов програмування.

Назва мови C# трактується як наступне покоління розвитку C++, а символ # – символізує “++++”. Спочатку в назві фігурував діз - “#”(англійською sharp), однак через відсутність цього символу на клавіатурі, використовується знак для позначення номеру “#”. Загалом назву мови можна позначати обома символами.

Розробка мови C# розпочалася в грудні 1998 року, та готувався до випуску разом з продуктами групи Millennium. Проект мав назву COOL(C-style Object Oriented Language), та розроблявся як аналог Java від компанії Oracle. C# був анонсований, широкому загалу, в 2000 році, як основна мова платформи Microsoft .NetFramework. В цьому ж році з'явилася перша загальнодоступна бета-версія.

Перша фінальна версія мови програмування C# була випущена в 2002 році разом з середовищем інтегрованої розробки програмного забезпечення VisualStudio .Net.

Подібно до Java, C# отримав наступні концепції:

- віртуальна машина – платформа .Net виконує програму подібно до віртуальної машини від Java;
- байт-код – програмний код компілюється в проміжкову мову MSIL(Microsoft Intermediate Language), а вже потім перетворюється на машинну мову, в залежності від платформи на якій запускається програма;
- керований код – оскільки програми, написані на C#, виконуються виключно у віртуальному середовищі CLR(Common Language Runtime), це дає змогу контролювати виконання програми та у будь який момент зупинити її, а також контролювати використання пам'яті програмою, за необхідності - збільшувати, чи видаляти частини пам'яті, які використовує програма.

Як Ви бачите, мова C# та платформа .Net постійно розвивається, вбираючи в себе найкращі практики використання мов програмування.

Окрім C#, середовище CLR підтримує багато інших мов програмування, зокрема в Microsoft розробляють VisualBasic .Net та F#. Крім цього існують .Net реалізації мов: Delphi.Net, IronPython, IronRuby, Boo, PascalABC.Net.

.Net дозволяє розробляти програмне забезпечення під різні операційні системи: Windows, MacOS, Linux, Android, iOS. Mono – реалізація .Net для Linux. Середовища для розробки: VisualStudio, Visual Studio Code, Unity3D, Xamarin, Sharp Develop, Mono Develop покривають всі операційні системи і технології, та дозволяють розробляти мобільні, настільні(desktop), Web, вбудовані(embedded) додатки та сервіси.

Середовище CLR та бібліотеки класів .NetFramework включають в себе основу для багатьох технологій, які розробники можуть використовувати при створенні продуктів. Зокрема LINQ, Entity Framework, ADO.Net – технології, що використовуються для роботи з даними. Технології Windows Forms, WPF, ASP.Net, ASP.Net MVC – для створення desktop та Web додатків.^[3]

Для зберігання даних використовується система баз даних SQLite. Компактна вбудована СУБД. Слово «вбудована» (embedded) означає, що SQLite не використовує парадигму клієнт-сервер, тобто движок SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а являє собою бібліотеку, з якої програма компонується, і движок стає складовою частиною програми. Таким чином, в якості протоколу обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, ключі та дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується програма. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції записи весь файл, який зберігає базу

даних, блокується. ACID-функції досягаються в тому числі за рахунок створення файлу журналу.

Кілька процесів або потоків можуть одночасно без будь-яких проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, якщо ніяких інших запитів в даний момент не обслуговується. В іншому випадку спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданного інтервалу часу. SQLite підтримує динамічне типизування даних. Зазначений при оголошенні поля, тип зберігається для довідки в його вихідному написанні, і використовується в якості основи для вибору переваг (так зване «typeaffinity»: це підхід, що рідко зустрічається в інших СУБД) при виконанні неявних перетворень типів на підставі схожості цієї назви типу на що-небудь, знайоме SQLite. Якщо безпечного перетворення значення в бажаний тип не виходить, SQLite записує значення в його початковому вигляді. Для отримання значень з бази є ряд функцій для кожного з типів, і якщо тип значення, що зберігається не відповідає запитуваній, воно теж, по можливості, перетворюється.

В SQLite підтримувані такі типи даних.

- NULL: NULL-значення. INTEGER : ціле зі знаком, що зберігається в 1, 2, 3, 4, 6, або 8 байтах.
- REAL: число з плаваючою комою, що зберігається в 8-байтовому форматі IEEE.
- TEXT: текстовий рядок з кодуванням UTF-8, UTF-16BE або UTF-16LE.
- BLOB: тип даних, що зберігається точно в такому ж вигляді, в якому і був отриманий.

Переваги SQLite.

- Файлова: вся база даних зберігається в одному файлі, що полегшує переміщення.
- Стандартизована: SQLite використовує SQL; деякі функції опущені (RIGHT OUTER JOIN або FOREACH STATEMENT), однак, є і деякі нові.
- Дуже добре підходить для розробки і навіть тестування: під час етапу розробки потребує масштабування. SQLite, зі своїм багатим набором функцій, може надати більш ніж достатній функціонал, при цьому будучи досить простий для роботи з одним файлом і пов'язаної бібліотекою.

Недоліки SQLite.

- Відсутність призначеного для користувача управління: просунуті БД надають користувачам можливість управляти зв'язками в таблицях відповідно до привілеями, але у SQLite такої функції немає.
- Неможливість додаткового налаштування: знову-таки, SQLite не можна зробити більш продуктивною, проходячи в налаштуванні - так вже вона влаштована.

Сама бібліотека SQLite написана на C; існує велика кількість прив'язок до інших мов програмування, в тому числі AppleSwift, Delphi, C++, Java, C#, VB.NET, Python, Perl, Node.js, PHP, PureBasic, Tcl (засоби для роботи з Tcl включені в комплект поставки SQLite), Ruby, Haskell, Scheme, Smalltalk, Lua і Parser, а також до багатьох інших.

Простота і зручність вбудовування SQLite привели до того, що бібліотека використовується в браузерах, музичних плеєрах і багатьох інших програмах.

Зокрема, SQLite використовується в:

Adobe Integrated Runtime - середовище для запуску додатків (частково);

Gears ;

Autoit ;

Фреймворк Qt ;

Платформа XUL на движку Gecko 1.9+, XULRunner 1.9+ і, потенційно, всі додатки, засновані на цій платформі, в тому числі:

Mozilla Firefox (починаючи з версії 3.0)

Skype ;

Viber ;

Деякі моделі GPS -навігаторів Garmin

Багато програм підтримують SQLite як формат зберігання даних (особливо в MacOS і iOS , Android), в тому числі:

1С: Підприємство 7.7;

1С: Підприємство 8.3;

Adobe Photoshop Lightroom ;

FlylinkDC ++ ;

AIMP ;

Banshee ;

Calibre ;

Eserv ;

F-Spot ;

Nextcloud ;

FARManager (починаючи з версії 3.0);

Gajim ;

Google Chrome ;

MyChat ;

Opera (починаючи з версії 10.50);

2.4. Опис структури системи та алгоритмів її функціонування

Інтерфейс даного додатку був розроблений в межах Visual Studio 2019, проектування інтерфейсу не потрібний для подібних програм, так як пріоритет в таких додатках є ефективність аналізу даних, які надходять з баз даних. Більшим пріоритетом в інтерфейсі були графіки зі статистикою.

Взаємодія з програмою відбувається через по одинарному кліку кнопок миші, що спрощує її взаємодію з користувачем.

Програма містить інформацію про всі продажі товарів в підприємстві та фінансову ефективність, що дозволяє визначати головні фактори ефективності та чинники великих витрат.

База даних про продажі та відвідувачів з покупцями приведена нижче.

| date | visitors | purchases | item1 | item2 | item3 | item4 | item5 |
|------------|----------|-----------|-------|-------|-------|-------|-------|
| 22.03.2022 | 179 | 96 | 3 | 3 | 50 | 40 | 0 |
| 23.03.2022 | 218 | 141 | 8 | 17 | 92 | 17 | 7 |
| 24.03.2022 | 124 | 92 | 7 | 20 | 12 | 27 | 26 |
| 25.03.2022 | 214 | 159 | 20 | 56 | 12 | 34 | 37 |
| 26.03.2022 | 167 | 107 | 8 | 26 | 56 | 7 | 10 |
| 28.03.2022 | 205 | 153 | 35 | 32 | 19 | 47 | 20 |
| 29.03.2022 | 135 | 91 | 21 | 25 | 8 | 33 | 4 |
| 30.03.2022 | 135 | 104 | 25 | 27 | 18 | 26 | 8 |
| 31.03.2022 | 186 | 131 | 29 | 47 | 15 | 22 | 18 |
| 01.04.2022 | 192 | 117 | 18 | 46 | 39 | 7 | 7 |
| 02.04.2022 | 150 | 96 | 9 | 38 | 2 | 12 | 35 |
| 03.04.2022 | 138 | 95 | 22 | 11 | 18 | 26 | 18 |
| 04.04.2022 | 136 | 116 | 11 | 38 | 6 | 43 | 18 |
| 05.04.2022 | 196 | 146 | 9 | 53 | 28 | 40 | 16 |
| 06.04.2022 | 207 | 140 | 14 | 24 | 56 | 15 | 31 |
| 07.04.2022 | 153 | 112 | 25 | 20 | 27 | 6 | 34 |
| 08.04.2022 | 127 | 110 | 21 | 3 | 10 | 47 | 29 |
| 09.04.2022 | 185 | 128 | 27 | 45 | 28 | 12 | 16 |
| 10.04.2022 | 169 | 128 | 23 | 7 | 8 | 28 | 62 |
| 11.04.2022 | 146 | 103 | 11 | 26 | 22 | 9 | 35 |

Рис. 2.1 – База даних про підприємство.

Відділи БД:

1. Date – відомість про день, місяць та рік, в який записались дані про товари та відвідувачів.
2. Visitors – інформація про людей, які зайшли в магазин, але нічого не купили, тобто сумарна кількість відвідувачів за день.
3. Purchases – кількість людей, які зайшли та зробили покупку в магазині.
4. Item1, item2, item3, item4, item5 – кількість продажу товару по групам в натуральних одиницях за день.

БД користувачів є окремою від бази даних показників. Створюється файл у форматі DAT, в якому міститься інформація про користувачів: ім'я користувача та зашифрований пароль.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

В якості вхідних програма використовує БД з показниками про дату, відвідувачів, покупців та груп товарів. Після натискання кнопки «Побудувати графік» проводиться розрахунок формул, зміна даних на графіках, множення одиниць товару на їх ціну та розрахунок прибутку приміщення, тобто прибуток на метр квадратний.

2.6. Опис роботи розробленої системи

2.6.1. Використані технічні засоби

Програма не є вимогливою до технічної складової, та може запускатись на ПК та ноутбуках під управлінням ОС Windows.

При розробці було використано ПК з наступними характеристиками:

Екран Samsung Dynamic Contrast 16” (1600x900), AMD Ryzen 5 3600 6-Core Processor 3.59 GHz, RAM 16 ГБ, SSD 1 ТБ, NVIDIA GeForce GTX 1050Ti 4 ГБ.

2.6.2. Використані програмні засоби

Розроблена в рамках кваліфікаційної роботи ПЗ було розроблено на мові C#, з використанням БД SQLite, в середовищі Visual Studio 2019.

2.6.3. Виклик та завантаження програми

Запуск програми є стандартним процесом exe-файлу, який може виконуватись на різних ПК, ноутбуках під управлінням ОС Windows.

Програма завантажується з архіву та після вилучення файлів з архіву на вказаний шлях на дисці. Після цього програма може функціонувати.

2.6.4. Опис інтерфейсу користувача

Програма має практичний та мінімалістичний інтерфейс, в якому переважну частину займають графіки та таблиця з бази даних. Вся інформація на головній сторінці зчитується після вказання періоду на таблиці та розрахунку через натискання кнопки «Побудувати графік». Зображення відділів програми наведено на рисунках 2.2 – 2.7

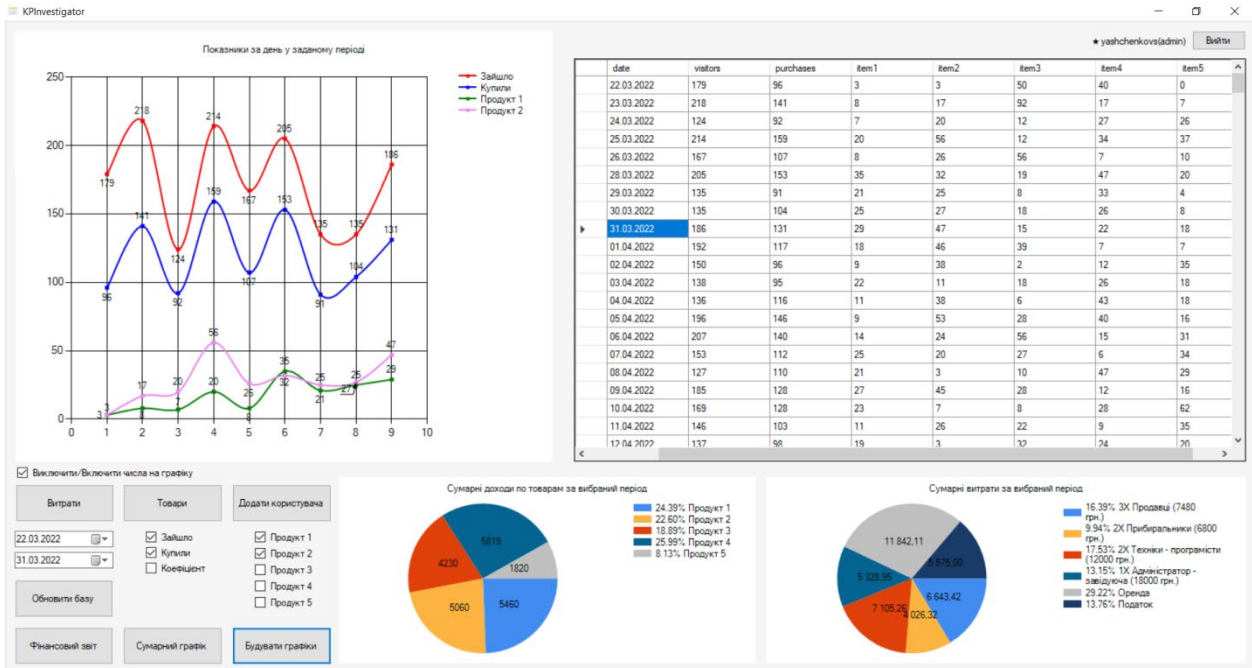


Рис.2.2 – Головний відділ

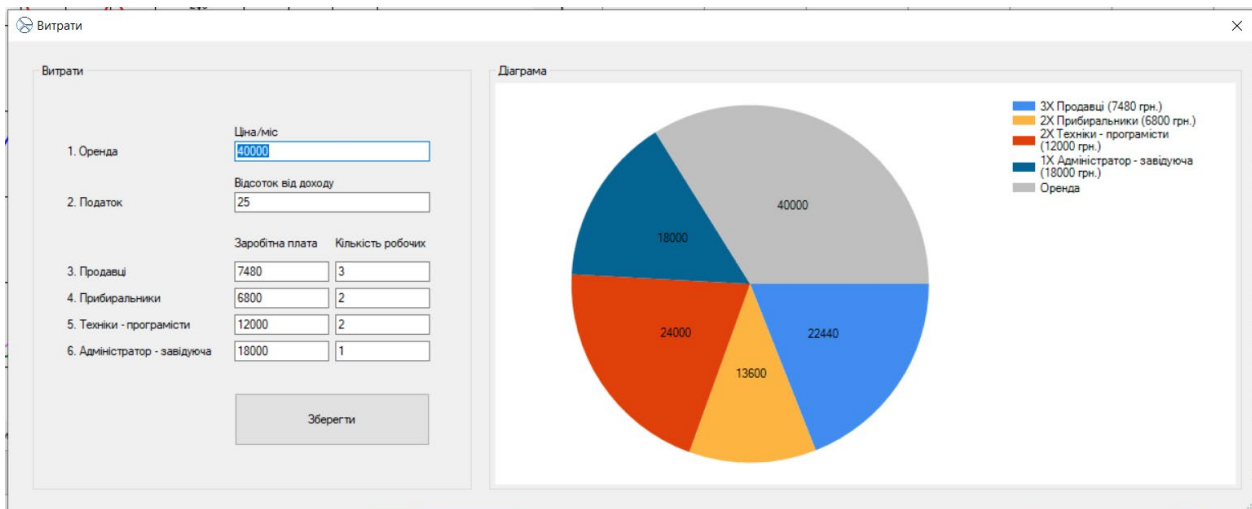


Рис.2.3 – Відділ витрат

Товари

| | Назва продукту | | Закупка | | Продажа |
|----|---|----|----------------------------------|----|----------------------------------|
| 1. | <input type="text" value="Протиалергенні"/> | 1. | <input type="text" value="100"/> | 1. | <input type="text" value="135"/> |
| 2. | <input type="text" value="Противірусні"/> | 2. | <input type="text" value="95"/> | 2. | <input type="text" value="115"/> |
| 3. | <input type="text" value="Септики"/> | 3. | <input type="text" value="60"/> | 3. | <input type="text" value="75"/> |
| 4. | <input type="text" value="Репеленти"/> | 4. | <input type="text" value="57"/> | 4. | <input type="text" value="80"/> |
| 5. | <input type="text" value="Перев'язувальні матері"/> | 5. | <input type="text" value="29"/> | 5. | <input type="text" value="43"/> |

Рис. 2.4 – Відділ товарів.

Користувачі

morozbi

Рис. 2.5 – відділ керування користувачами.

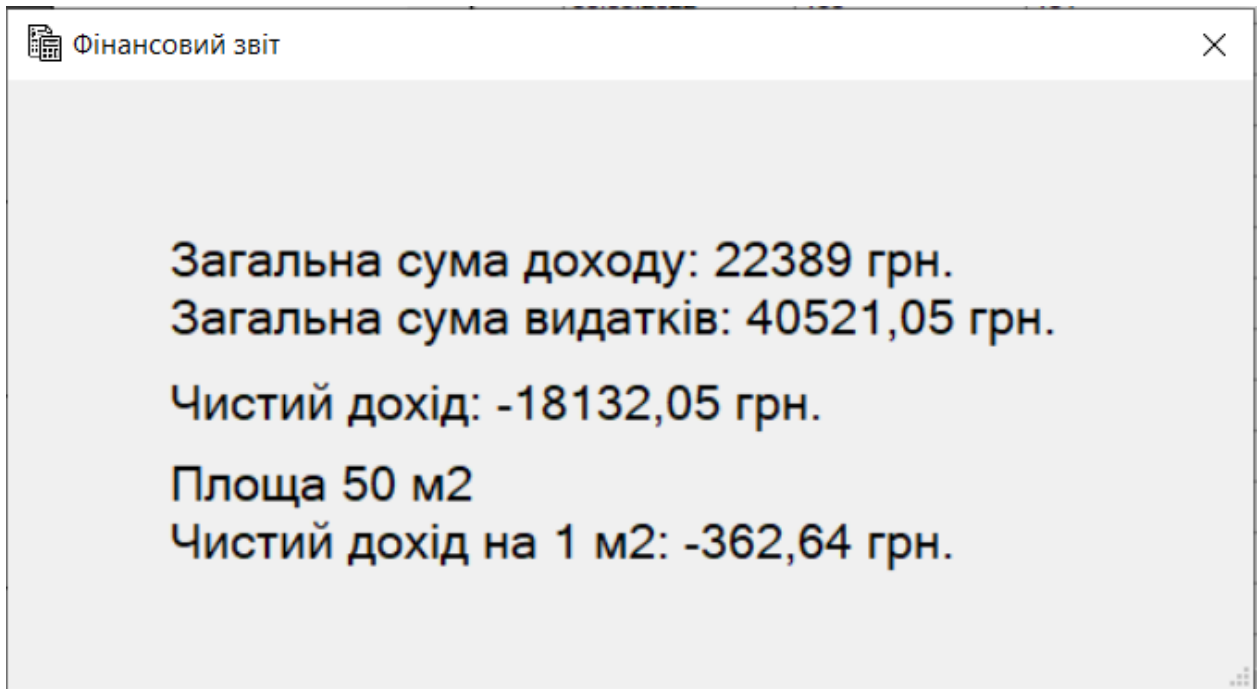


Рис. 2.6 – відділ фінансового звіту.

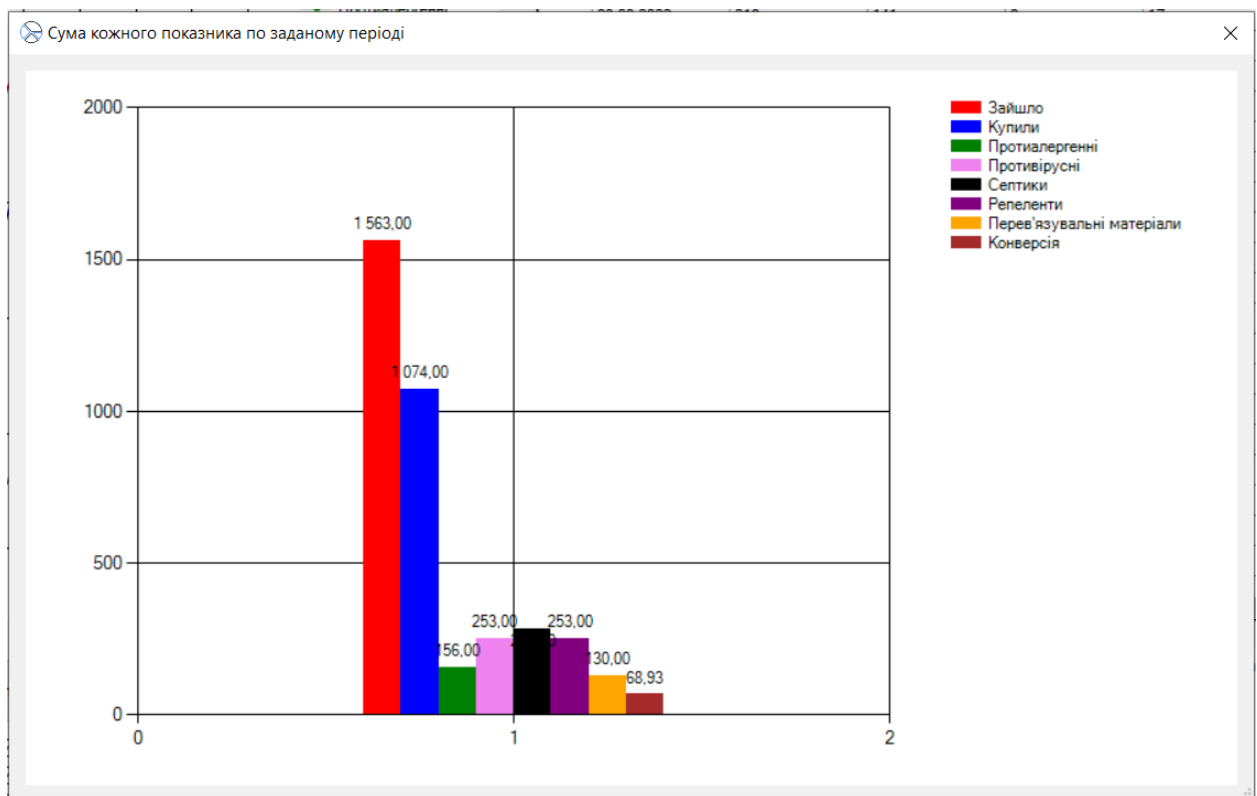


Рис.2.6 – відділ сумарного графіку на заданий період.

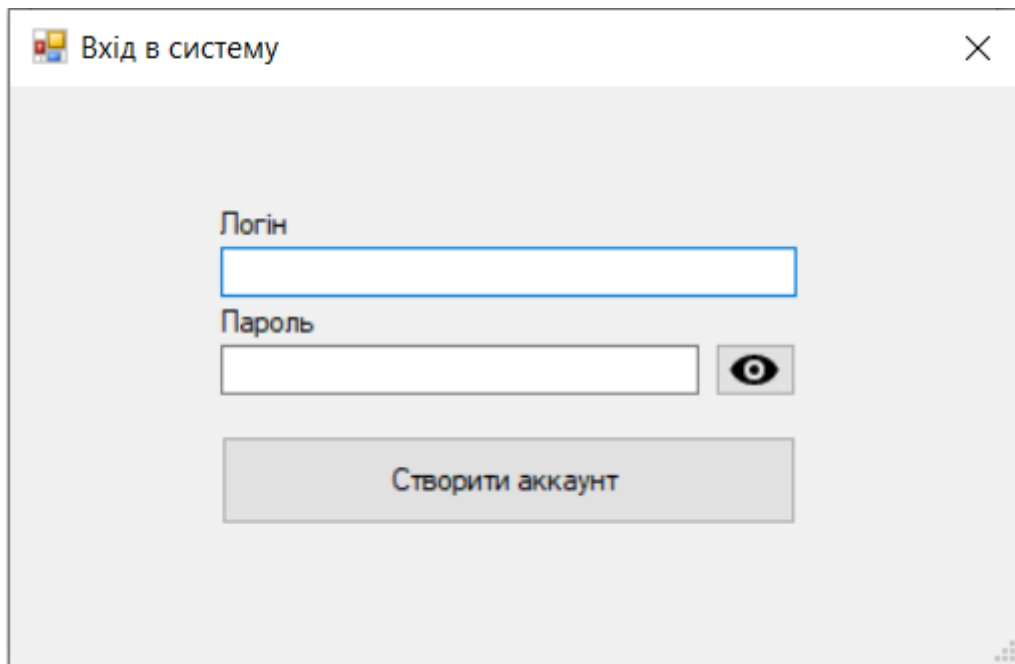


Рис.2.7 – вікно входу в програму.

РОЗДІЛ 3 ЕКОНОМІКА

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 950;
2. коефіцієнт складності програми – 1,8;
3. коефіцієнт корекції програми в ході її розробки – 0,09;
4. година заробітна плата програміста – 90 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
7. вартість машино-години ЕОМ – 15 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{omл} + t_{\partial}, \text{ людино-годин, (4.1)}$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{omл}$ - витрати праці на налагодження програми на ЕОМ;

t_{∂} - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p)$$

де q - передбачуване число операторів (950);

C - коефіцієнт складності програми (2,6);

p - коефіцієнт корекції програми в ході її розробки (0,09).

Звідси умовне число операторів:

$$Q = q \cdot C \cdot (1 + p) = 1,8 \cdot 950 \cdot (1+0,09) = 1863$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин,}$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає 1,2.

Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1.2$). З урахуванням коефіцієнта кваліфікації $k = 1,2$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (1863 \cdot 1,2) / (75 \cdot 1,2) = 2236 / 90 = 24,85 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин, (3.2)}$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.2), отримаємо:

$$t_a = 1863 / (20 \cdot 1,2) = 77 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин.}$$

$$t_n = 1863 / (25 \cdot 1,2) = 62,1 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ люДИНО-ГОДИН.}$$

$$t_{oml} = 1863 / (5 \cdot 1,2) = 310 \text{ люДИНО-ГОДИН.}$$

- за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ люДИНО-ГОДИН.}$$

$$t_{oml}^k = 1,5 \cdot 310 = 465 \text{ люДИНО-ГОДИН.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{dp} + t_{do}, \text{ люДИНО-ГОДИН,}$$

де t_{dp} - трудомісткість підготовки матеріалів і рукопису:

$$t_{dp} = \frac{Q}{(15..20) \cdot k}, \text{ люДИНО-ГОДИН,}$$

t_{do} - трудомісткість редагування, печатки й оформлення документації:

$$t_{do} = 0,75 \cdot t_{dp}, \text{ люДИНО-ГОДИН.}$$

Підставляючи відповідні значення, отримаємо:

$$t_{dp} = 1863 / (18 \cdot 1,2) = 86,25 \text{ люДИНО-ГОДИН.}$$

$$t_{do} = 0,75 \cdot 86,25 = 64,68 \text{ люДИНО-ГОДИН.}$$

$$t_d = 86,25 + 70 = 156,25 \text{ люДИНО-ГОДИН.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 24,85 + 77 + 62,1 + 310 + 156,25 = 680,2 \text{ людино-годин.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{ПО}$ включають витрати на заробітну плату виконавця програми $Z_{ЗП}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ЗП}, \text{ грн,}$$

де: t - загальна трудомісткість, людино-годин;

$C_{ЗП}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 120 грн / год, отримуємо:

$$Z_{ЗП} = 680,2 \cdot 90 = 61\,218 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн, (3.3)}$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (13 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$З_{мв} = 310 \cdot 15 = 4650 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 61\,218 + 4650 = 65\,868 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.}$$

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Звідси витрати на створення програмного продукту:

$$T = 680,2 / 1 \cdot 176 = 3,8 \approx 4 \text{ міс.}$$

Висновок. Програмне забезпечення призначене для аналізу даних. Вартість даного програмного забезпечення становить 65 868 грн. і не вимагає додаткових витрат як при розробці програми. Очікуваний час розробки становить 4 місяці. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого

завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

ВИСНОВКИ

Створення затребуваного продукту у сфері аналізу даних комерційного підприємства є відповідальним та трудомістким процесом, у якому важлива участь самого користувача програмного забезпечення під час розробки для уточнювання та опису функцій, які необхідні для роботи з даними.

Для аналізу вхідних даних було створено БД усіх необхідних показників підприємства, були прописані формули розрахунку та графіки для аналізу, відображення програми на стороні та була налаштована взаємодія користувача з ними.

Даний додаток відповідає всім нинішнім стандартам створення програмного забезпечення й надає усі необхідні функції, які були узгодженні з навчальним керівником у процесі розробки.

Розроблене у межах кваліфікаційної роботи програмне забезпечення було реалізоване на мові C# з використанням БД SQLite у середовищі Visual Studio 2019.

Всі поставлені в роботі завдання виконано в повному обсязі.

Також у кваліфікаційній роботі було визначено трудомісткість розробленої системи (680, 2 люд-год), проведений підрахунок вартості роботи по створенню програми (65 868 грн) та розраховано час на її створення (4 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Науковий журнал Комп'ютерно – Інтегровані технології: освіта, наука, виробництво.
Automated System for Analyzing and Researching the effectiveness of Business operations. В. Moroz, O.Syrotkina, A.Marochko. 2019, с. 55.
Електронне джерело:
https://scholar.google.com.ua/citations?view_op=view_citation&hl=ru&user=i5B7-AMAAAAJ&cstart=20&pagesize=80&sortby=pubdate&citation_for_view=i5B7-AMAAAAJ:QUX0mv85b1cC
2. https://pidru4niki.com/1629102250032/marketing/rozdrubna_torgivlya_sut_funktsiyi_vidi
3. Вступ в С#. Посібник
<https://programm.top/uk/c-sharp/tutorial/introduction/>
4. Реляційні бази даних
https://pns.hneu.edu.ua/pluginfile.php/601514/mod_resource/content/1/Лекція%2003%20Реляційні%20бази%20даних.pdf
5. Розуміння SQLite
<https://products.aspose.com/gis/uk/net/gis-formats/sqlite/>
6. Методології розробки програмного забезпечення, його життєвий цикл.
<https://evergreens.com.ua/ua/articles/software-development-metodologies.html>
7. Оцінка ефективності торговельного підприємства

- https://pidru4niki.com/14051003/ekonomika/_otsinka_efektivnosti_diyalnosti_torgovelnogo_pidpriyemstva
8. Вступ до об'єктно – орієнтованого програмування C#
https://dut.edu.ua/uploads/1_1355_77924598.pdf
 9. Уроки для вивчення C#
<https://programer.in.ua/index.php/uroky/uroky-dlia-vyvchennia-c>
 10. Список і опис типів даних SQLite
<https://ciksiti.com/uk/chapters/9013-sqlite-data-types-list-and-description>
 11. Роздрібна торгівля, її суть, функції та види
https://pidru4niki.com/1629102250032/marketing/rozdribna_torgivlya_sut_funktsiyi_vidi
 12. Ефективність роботи роздрібного магазину.
<https://abmcloud.com/efektivnist-roboti-rozdribnogo-magazinu/>
 13. Методичні вказівки з виконання економічного розділу у дипломних проектах студентів спеціальності «Комп'ютерні системи» / О.Г. Вагонова, О.Б. Нікітіна, Н.М. Романюк; М-во освіти та науки України, ДВНЗ «Нац. гірн. ун-т». - Д.: НГУ, 2013. - 11 с.
 14. Введення в розробку програмного забезпечення.
<https://acode.com.ua/urok-3-vvedennya-v-rozrobku-programnogo-zabezpechennya/>
 15. Аналіз даних
<https://uk.economy-pedia.com/11033079-analysis-of-data>
 16. Основи статистики та аналізу даних
<https://socialdata.org.ua/manual/manual4/>
 17. Реалізація плану дослідження
https://pidru4niki.com/1739061741719/marketing/realizatsiya_planu_doslidzhennya
 18. Маркетинговий аналіз ринку
<https://pro-consulting.ua/ua/pressroom/marketingovyj-analiz-rynka-instrukciya-po-primeneniyu>

19. Сутність маркетингового аналізу

https://pidru4niki.com/12631113/ekonomika/marketingoviy_analiz

20. Маркетинговий аналіз ринку: що потрібно знати підприємцю?

<https://torgsoft.ua/articles/stati/marketingovyij-analiz-ryinka-chto-nuzhno-znat-predprinimatelyu/>

ДОДАТОК А

КОДПРОГРАМИ

```
using system;
using system.collections.generic;
using system.componentmodel;
using system.data;
using system.data.sqlite;
using system.drawing;
using system.linq;
using system.text;
using system.threading.tasks;
using system.windows.forms;
using system.windows.forms.datavisualization.charting;
using system.io;
using system.globalization;

namespace diagram
{
    public partial class form1 : form
    {
        string dohod = "";
        string rashod = "";
        string clear_dohod = "";
    }
}
```

```

string m2 = "";

private sqliteconnection connection;

contextmenustrip contextmenu = new contextmenustrip();
toolstripmenuItem minimummenuItem = new toolstripmenuItem("мінімум");
toolstripmenuItem maximummenuItem = new toolstripmenuItem("максимум");
int zakupka1 = 0, zakupka2 = 0, zakupka3 = 0, zakupka4 = 0, zakupka5 = 0,
prodaja1 = 0, prodaja2 = 0, prodaja3 = 0, prodaja4 = 0, prodaja5 = 0;

int id_start = 0;
int id_end = 0;
int select_id = 0;

float global_4_visitors = 0;
float global_4_purchases = 0;
float global_4_item1 = 0;
float global_4_item2 = 0;
float global_4_item3 = 0;
float global_4_item4 = 0;
float global_4_item5 = 0;
float global_4_conversion = 0;

string name_product1="";
string name_product2 = "";
string name_product3 = "";
string name_product4 = "";
string name_product5 = "";

public form1()
{
    initializecomponent();

    try
    {
        string[] lines = file.readalllines("data1.txt");
        zakupka1 = int32.parse(lines[0]);
        zakupka2 = int32.parse(lines[1]);
        zakupka3 = int32.parse(lines[2]);
        zakupka4 = int32.parse(lines[3]);
        zakupka5 = int32.parse(lines[4]);
        prodaja1 = int32.parse(lines[5]);
        prodaja2 = int32.parse(lines[6]);
        prodaja3 = int32.parse(lines[7]);
        prodaja4 = int32.parse(lines[8]);
        prodaja5 = int32.parse(lines[9]);
    }
}

```



```

name_product1 =lines[10].toString();
name_product2 = lines[11].toString();
name_product3 = lines[12].toString();
name_product4 = lines[13].toString();
name_product5 = lines[14].toString();

checkbox3.text = name_product1;
checkbox4.text = name_product2;
checkbox5.text = name_product3;
checkbox7.text = name_product4;
checkbox6.text = name_product5;
}
catch (exception)
{
}
contextmenu.items.add(minimummenuItem);
contextmenu.items.add(maximummenuItem);
datagridview1.contextmenustrip = contextmenu;

minimummenuItem.click += minimummenuItem_click;
maximummenuItem.click += maximummenuItem_click;

combobox1.selectedIndex = 0;

connection = new sqliteconnection("data source=" + directory.getcurrentdirectory()
+ "/mydatabase.db");

connection.open();

sqlitecommand command = new sqlitecommand("select * from data", connection);
sqlitedataadapter adapter = new sqlitedataadapter(command);
datatable table = new datatable();
adapter.fill(table);
datagridview1.datasource = table;
connection.close();

chart1.series[0].points.add(20);
chart1.series[0].points.add(30);
chart1.series[0].points.add(50);

chart2.series[0].points.add(20);
chart2.series[0].points.add(30);
chart2.series[0].points.add(50);

chart3.series[0].points.add(20);
chart3.series[0].points.add(30);

```

```

chart3.series[0].points.add(50);

chart1.series[0].points[0].color = color.red;
chart1.series[0].points[1].color = color.blue;
chart1.series[0].points[2].color = color.green;

chart2.series[0].points[0].color = color.red;
chart2.series[0].points[1].color = color.blue;
chart2.series[0].points[2].color = color.green;

chart3.series[0].points[0].color = color.red;
chart3.series[0].points[1].color = color.blue;
chart3.series[0].points[2].color = color.green;

chart1.series[0].charttype = seriescharttype.pie;
chart3.series[0].charttype = seriescharttype.doughnut;
chart4.series[0].charttype = seriescharttype.rangebar;

form4 form4 = new form4();
form4.startposition = formstartposition.centerparent;
form4.showdialog();

show();
chart2.titles.clear();
chart2.titles.add("показники за день у заданому періоді");
chart4.titles.clear();
chart4.titles.add("сумарні показники за вибраний період");
chart5.titles.clear();
chart5.titles.add("сумарні доходи по товарам за вибраний період");
chart6.titles.clear();
chart6.titles.add("сумарні витрати за вибраний період");

string user_type = form4.usertype;
if (user_type != "off")
{
    if (user_type == "admin")
    {

        label1.text = "★ " + form4.username + "(" + user_type + ")";

        tooltip1.settooltip(label1, "адмін може створювати нових користувачів.");

        button1.visible = true;
    }
}

```

```

else
{
    label1.text = form4.username + "(" + user_type + ")";

    tooltip1.settooltip(label1, "користувач може бачити данні програми та
змінювати їх.");

    button1.visible = false;

}
label1.location = new point(label1.location.x - label1.width+40,
label1.location.y);

}
else
{
    application.exit();
}

button3.performclick();
this.windowstate = formwindowstate.maximized;

}
private void button2_click(object sender, EventArgs e)
{
    connection.open();

    using (var reader = new StreamReader("load_file.txt"))
    {
        try
        {
            using (var command = new SqlCommand(connection))
            {
                command.CommandText = "delete from data";
                command.ExecuteNonQuery();
            }

            string line;
            while ((line = reader.ReadLine()) != null)
            {
                string[] parts = line.Split('|');

                using (var command = new SqlCommand("insert into data (date, visitors,
purchases, item1, item2, item3, item4, item5) values (@date, @visitors, @purchases,
@item1, @item2, @item3, @item4, @item5)", connection))
                {

```

```

        string date_my = parts[0];
        datetime date = datetime.parseexact(date_my, "dd.mm.yyyy",
cultureinfo.invariantculture);
        string date_formatted = date.tostring("yyyy-mm-dd");

        command.parameters.addwithvalue("@date", date_formatted);
        command.parameters.addwithvalue("@visitors", int.parse(parts[1]));
        command.parameters.addwithvalue("@purchases", int.parse(parts[2]));
        command.parameters.addwithvalue("@item1", int.parse(parts[3]));
        command.parameters.addwithvalue("@item2", int.parse(parts[4]));
        command.parameters.addwithvalue("@item3", int.parse(parts[5]));
        command.parameters.addwithvalue("@item4", int.parse(parts[6]));
        command.parameters.addwithvalue("@item5", int.parse(parts[7]));
        command.executenonquery();
    }

}
}
catch (exception ex)
{
    messagebox.show(ex.message);
}

connection.close();
}
}

private void button1_click_1(object sender, EventArgs e)
{
    connection.open();

    chart2.series[0].points.clear();

    using (var command = new SqlCommand("select * from data", connection))
    {
        using (var reader1 = command.ExecuteReader())
        {
            while (reader1.Read())
            {
                string date = reader1.GetString(0);
                int visitors = reader1.GetInt32(1);
                int purchases = reader1.GetInt32(2);
                int item1 = reader1.GetInt32(3);
                int item2 = reader1.GetInt32(4);
                int item3 = reader1.GetInt32(5);
                int item4 = reader1.GetInt32(6);
                int item5 = reader1.GetInt32(7);
            }
        }
    }
}

```

```

        chart2.series[0].points.add(visitors);
    }
}
connection.close();
}

private void button3_click(object sender, EventArgs e)
{

    string datefromstr = dateTimePicker1.Value.Date.ToString("yyyy-mm-dd");
    string datetostr = dateTimePicker2.Value.Date.ToString("yyyy-mm-dd");

    DateTime datefrom = DateTime.ParseExact(datefromstr, "yyyy-mm-dd",
cultureinfo.InvariantCulture);
    DateTime dateto = DateTime.ParseExact(datetostr, "yyyy-mm-dd",
cultureinfo.InvariantCulture);

    if (datefrom.CompareTo(dateto) <= 0)
    {
        // продовжаєм роботу програми
        program_grafic(datefromstr, datetostr);
    }
    else
    {
        MessageBox.Show("не коректно вибрано дати");
    }
}

public void program_grafic(string datefromstr, string datetostr)
{

    chart5.series[0].points.clear();
    chart6.series[0].points.clear();

    chart2.series.clear();
    chart2.series.add("зайшло");
    chart2.series.add("купили");
    chart2.series.add(name_product1);
    chart2.series.add(name_product2);
    chart2.series.add(name_product3);
    chart2.series.add(name_product4);
    chart2.series.add(name_product5);
    chart2.series.add("конверсія");

    chart4.series.clear();
    chart4.series.add("зайшло");
    chart4.series.add("купили");

```

```

chart4.series.add(name_product1);
chart4.series.add(name_product2);
chart4.series.add(name_product3);
chart4.series.add(name_product4);
chart4.series.add(name_product5);
chart4.series.add("конверсія");

chart2.series[0].color = color.red;
chart2.series[1].color = color.blue;
chart2.series[2].color = color.green;
chart2.series[3].color = color.violet;
chart2.series[4].color = color.black;
chart2.series[5].color = color.purple;
chart2.series[6].color = color.orange;
chart2.series[7].color = color.brown;

chart4.series[0].color = color.red;
chart4.series[1].color = color.blue;
chart4.series[2].color = color.green;
chart4.series[3].color = color.violet;
chart4.series[4].color = color.black;
chart4.series[5].color = color.purple;
chart4.series[6].color = color.orange;
chart4.series[7].color = color.brown;

for (int i = 0; i < 8; i++)
{
    chart4.series[i].points.clear();
}

for (int i = 0; i < 8; i++)
{
    chart2.series[i].points.clear();
    chart2.series[i].isvisibleinlegend = false;
}

for (int i = 0; i < 8; i++)
{
    chart2.series[i].borderwidth = 2;
    chart2.series[i].charttype = seriescharttype.spline;
    chart2.series[i].markerstyle = markerstyle.square;
}

if (combobox1.selectedindex == 0)//Дні
{
    string query = "select * from data where date between date(@datefrom) and
date(@dateto)";

```

```

connection.open();
int visitors = 0;
int purchases = 0;

int item1 = 0;
int item2 = 0;
int item3 = 0;
int item4 = 0;
int item5 = 0;

float konvertion = 0;

using (sqlitecommand command = new sqlitecommand(query, connection))
{
    command.parameters.addwithvalue("@datefrom", datefromstr);
    command.parameters.addwithvalue("@dateto", datetostr);

    datetime date1 = datetime.parse(datefromstr);
    datetime date2 = datetime.parse(datetostr);

    timespan difference = date2 - date1;
    int days = difference.days;
    id_start = 0;
    id_end = days;

    using (sqlitedatareader reader = command.executereader())
    {

        int count = 0;
        while (reader.read())
        {
            try
            {
                count++;

                visitors += convert.toint32(reader["visitors"]);
                purchases += convert.toint32(reader["purchases"]);

                int visitors0 = convert.toint32(reader["visitors"]);
                int purchases0 = convert.toint32(reader["purchases"]);

                item1 += convert.toint32(reader["item1"]);
                item2 += convert.toint32(reader["item2"]);
                item3 += convert.toint32(reader["item3"]);
                item4 += convert.toint32(reader["item4"]);
                item5 += convert.toint32(reader["item5"]);
            }
        }
    }
}

```

```

float rez = (float)purchases0 / visitors0;
konversion += rez * 100;

if (checkbox1.checked == true)
{
    chart2.series[0].points.addxy(count,
convert.toint32(reader["visitors"]));
    chart2.series[0].isvisibleinlegend = true;
}
if (checkbox2.checked == true)
{
    chart2.series[1].points.addxy(count,
convert.toint32(reader["purchases"]));
    chart2.series[1].isvisibleinlegend = true;
}
if (checkbox3.checked == true)
{
    chart2.series[2].points.addxy(count,
convert.toint32(reader["item1"]));
    chart2.series[2].isvisibleinlegend = true;
}
if (checkbox4.checked == true)
{
    chart2.series[3].points.addxy(count,
convert.toint32(reader["item2"]));
    chart2.series[3].isvisibleinlegend = true;
}
if (checkbox5.checked == true)
{
    chart2.series[4].points.addxy(count,
convert.toint32(reader["item3"]));
    chart2.series[4].isvisibleinlegend = true;
}
if (checkbox7.checked == true)
{
    chart2.series[5].points.addxy(count,
convert.toint32(reader["item4"]));
    chart2.series[5].isvisibleinlegend = true;
}
if (checkbox6.checked == true)
{
    chart2.series[6].points.addxy(count,
convert.toint32(reader["item5"]));
    chart2.series[6].isvisibleinlegend = true;
}
if (checkbox8.checked == true)
{
    float raz_fl = (float)purchases0 / visitors0;

```



```

        chart2.series[7].points.addxy(count, raz_fl * 100);
        chart2.series[7].isvisibleinlegend = true;
    }

    for (int n = 0; n < chart2.series[7].points.count; n++)
    {
        chart2.series[7].points[n].labelformat = "#,###0.00";
    }

}
catch (exception ex)
{
}

}

chart4.series[0].points.add(visitors);
chart4.series[1].points.add(purchases);
chart4.series[2].points.add(item1);
chart4.series[3].points.add(item2);
chart4.series[4].points.add(item3);
chart4.series[5].points.add(item4);
chart4.series[6].points.add(item5);

global_4_visitors = visitors;
global_4_purchases = purchases;
global_4_item1 = item1;
global_4_item2 = item2;
global_4_item3 = item3;
global_4_item4 = item4;
global_4_item5 = item5;

int dohod1 = (item1 * prodaja1) - (item1 * zakupka1);
int dohod2 = (item2 * prodaja2) - (item2 * zakupka2);
int dohod3 = (item3 * prodaja3) - (item3 * zakupka3);
int dohod4 = (item4 * prodaja4) - (item4 * zakupka4);
int dohod5 = (item5 * prodaja5) - (item5 * zakupka5);

float all_suma= dohod1 + dohod2 + dohod3 + dohod4 + dohod5;
float one_procent = (all_suma / 100);

float procent_1 = dohod1 / one_procent;
float procent_2 = dohod2 / one_procent;
float procent_3 = dohod3 / one_procent;
float procent_4 = dohod4 / one_procent;
float procent_5 = dohod5 / one_procent;

chart5.series[0].points.addxy(procent_1.toString("f2",

```

```

cultureinfo.invariantculture) + "% " + name_product1, dohod1);
    chart5.series[0].points.addxy(procent_2.toString("f2",
cultureinfo.invariantculture) + "% " + name_product2, dohod2);
    chart5.series[0].points.addxy(procent_3.toString("f2",
cultureinfo.invariantculture) + "% " + name_product3, dohod3);
    chart5.series[0].points.addxy(procent_4.toString("f2",
cultureinfo.invariantculture) + "% " + name_product4, dohod4);
    chart5.series[0].points.addxy(procent_5.toString("f2",
cultureinfo.invariantculture) + "% " + name_product5, dohod5);

int all_dodhod = dohod1 + dohod2 + dohod3 + dohod4 + dohod5;
dohod = "загальна сума доходу: " + all_dodhod + " грн.";

chart5.series[0].borderwidth = 2;
chart5.series[0].charttype = seriescharttype.spline;
chart5.series[0].markerstyle = markerstyle.square;
chart5.series[0].color = color.red;
chart5.series[0].charttype = seriescharttype.pie;

for (int i = 0; i < 5; i++)
{
    chart5.series[0].points[i].isvalueshownaslabel = true;
}

try
{
    using (streamreader reader3 = new streamreader("data.txt"))
    {
        // textbox1 - orenda
        int orenda = int32.parse(reader3.readline());

        // textbox6 - nalog
        int nalog = int32.parse(reader3.readline());

        // textbox8 - worker_grop1
        int worker_grop1 = int32.parse(reader3.readline());

        // textbox7 - worker_count1
        int worker_count1 = int32.parse(reader3.readline());

        // textbox10 - worker_grop2
        int worker_grop2 = int32.parse(reader3.readline());

        // textbox9 - worker_count2
        int worker_count2 = int32.parse(reader3.readline());

        // textbox12 - worker_grop3
        int worker_grop3 = int32.parse(reader3.readline());
    }
}

```

```

// textbox11 - worker_count3
int worker_count3 = int32.parse(reader3.readline());

// textbox3 - worker_grop4
int worker_grop4 = int32.parse(reader3.readline());

// textbox2 - worker_count4
int worker_count4 = int32.parse(reader3.readline());

int count_day = id_end - id_start;
float val1 = (worker_grop1 * worker_count1 / 30.4f) * count_day;
float val2 = (worker_grop2 * worker_count2 / 30.4f) * count_day;
float val3 = (worker_grop3 * worker_count3 / 30.4f) * count_day;
float val4 = (worker_grop4 * worker_count4 / 30.4f) * count_day;
float val5 = (orenda / 30.4f) * count_day;

float real_nalog = (all_dodhod / 100) * nalog;
float all_rashod = (val1 + val2 + val3 + val4 + val5 + real_nalog);

rashod = "загальна сума видатків: " + all_rashod + " грн.";

float clear_dodhod = all_dodhod - all_rashod;

clear_dohod = "чистий дохід: " + clear_dodhod + " грн.";

double result = clear_dodhod / 50;
result = math.round(result, 2);
string formattedresult = string.format("{0:n2}", result);

m2 = "площа 50 м2 \r\nчистий дохід на 1 м2: " + formattedresult + "
грн.";

float all_suma6 = val1+ val2+ val3+ val4+ val5+ real_nalog;
float one_procent6 = (all_suma6 / 100);

float procent6_1 = val1 / one_procent6;
float procent6_2 = val2 / one_procent6;
float procent6_3 = val3 / one_procent6;
float procent6_4 = val4 / one_procent6;
float procent6_5 = val5 / one_procent6;
float procent6_6 = real_nalog / one_procent6;

chart6.series[0].points.addxy(procent6_1.toString("f2",
cultureinfo.invariantculture) + "% " + worker_count1 + "х продавці (" + worker_grop1 + "
грн.)", val1);
chart6.series[0].points.addxy(procent6_2.toString("f2",

```

```

cultureinfo.invariantculture) + "% " + worker_count2 + "х прибиральники (" +
worker_grop2 + " грн.)", val2);
        chart6.series[0].points.addxy(procent6_3.toString("f2",
cultureinfo.invariantculture) + "% " + worker_count3 + "х техніки - програмісти (" +
worker_grop3 + " грн.)", val3);
        chart6.series[0].points.addxy(procent6_4.toString("f2",
cultureinfo.invariantculture) + "% " + worker_count4 + "х адміністратор - завідуюча (" +
worker_grop4 + " грн.)", val4);
        chart6.series[0].points.addxy(procent6_5.toString("f2",
cultureinfo.invariantculture) + "% " + "оренда", val5);
        chart6.series[0].points.addxy(procent6_6.toString("f2",
cultureinfo.invariantculture) + "% " + "податок", real_nalog);
    }
}
catch (exception)
{
}

chart6.series[0].borderwidth = 2;
chart6.series[0].charttype = seriescharttype.spline;
chart6.series[0].markerstyle = markerstyle.square;
chart6.series[0].color = color.red;

chart6.series[0].charttype = seriescharttype.pie;
for (int i = 0; i < 6; i++)
{
    chart6.series[0].points[i].isvalueshownaslabel = true;
    chart6.series[0].points[i].labelformat = "#,##0.00";
}

float    konvertion1    =    (float)math.round(konvertion,    2,
midpointrounding.awayfromzero);
chart4.series[7].points.add(konvertion1 / count);
global_4_conversion = konvertion1 / count;

for (int i = 0; i < 9; i++)
{

    try
    {

        chart4.series[i].points[0].isvalueshownaslabel = true;

    }
    catch (exception ex)
    { }

    if (checkbox9.checked == true)

```

```

        {
            try
            {
                for (int r = 0; r < chart2.series[i].points.count; r++)
                {
                    chart2.series[i].points[r].isvalueshownaslabel = true;
                }
            }
            catch (exception ex)
            {}
        }
        if (checkbox9.checked == false)
        {
            try
            {
                for (int r = 0; r < chart2.series[i].points.count; r++)
                {
                    chart2.series[i].points[r].isvalueshownaslabel = false;
                }
            }
            catch (exception ex)
            {}
        }
    }

    chart4.series[7].points[0].labelformat = "#,##0.00";
}
}
connection.close();
}
}

private void datagridview1_selectionchanged(object sender, eventargs e)
{
    if (datagridview1.selectedrows.count > 0)
    {
        datagridviewrow row = datagridview1.selectedrows[0];
        string value = row.cells[0].value.tostring();
        select_id = int32.parse( value);
    }
}

private void datagridview1_cellendedit(object sender, datagridviewcelleventargs e)
{
    int rowindex = e.rowindex;
    int columnindex = e.columnindex;
    datagridviewcell editedcell = datagridview1.rows[rowindex].cells[columnindex];
    string newvalue = editedcell.value.tostring();
}

```

```

if (newvalue.indexOf(" 0:00:00") != -1)
{
    newvalue = newvalue.replace(" 0:00:00", "");
    string[] mass_date = newvalue.split('.');
    string day = mass_date[0];
    string month = mass_date[1];
    string year = mass_date[2];
    newvalue = year + "-" + month + "-" + day;
}

int id = convert.ToInt32(datagridview1.rows[rowindex].cells[0].value);

datagridviewcolumn column = datagridview1.columns[columnindex];

string columnname = column.name;

connection.open();
string query = "update data set "+ columnname + " = @newvalue where id =
@id";
using (sqlitecommand command = new sqlitecommand(query, connection))
{
    command.parameters.addwithvalue("@newvalue", newvalue);
    command.parameters.addwithvalue("@id", id);
    command.executenonquery();
}
connection.close();

}

private void datagridview1_cellmouseclick(object sender,
datagridviewcellmouseeventargs e)
{
    if (datagridview1.selectedrows.count > 0)
    {
        datagridviewrow row = datagridview1.selectedrows[0];
        string value = row.cells[0].value.toString();
        int columnindex = datagridview1.currentcell.columnindex;
    }
}

private void minimummenuItem_click(object sender, eventargs e)
{
    datagridviewcell cell = datagridview1.selectedcells[0];
    int columnindex = cell.columnindex;
    string value = datagridview1.rows[cell.rowindex].cells[1].value.toString();
    string date = datagridview1.rows[cell.rowindex].cells[1].value.toString();
    textbox2.text = date;
}

```

```

datetimepicker1.text = date;

id_start = int32.parse(datagridview1.rows[cell.rowindex].cells[0].value.toString());

messagebox.show(value);
}

private void maximummenuItem_click(object sender, EventArgs e)
{
    datagridviewcell cell = datagridview1.selectedcells[0];
    int columnIndex = cell.columnindex;
    string value = datagridview1.rows[cell.rowindex].cells[1].value.toString();
    string date = datagridview1.rows[cell.rowindex].cells[1].value.toString();
    textbox3.text = date;
    datetimepicker2.text = date;
    id_end = int32.parse( datagridview1.rows[cell.rowindex].cells[0].value.toString());
    messagebox.show(value);
}

private void datagridview1_cellclick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        datagridviewrow row = datagridview1.rows[e.rowindex];
        string value = row.cells[0].value.toString();
        select_id = int32.parse(value);
    }
    catch (Exception ex)
    { }
}

private void button4_click_1(object sender, EventArgs e)
{
    form2 form2 = new form2();
    form2.startposition = formstartposition.centerparent;
    form2.showdialog();
}

private void button5_click(object sender, EventArgs e)
{
    form3 form3 = new form3();
    form3.startposition = formstartposition.centerparent;
    form3.showdialog();

    try
    {
        string[] lines = file.readalllines("data1.txt");
        zakupka1 = int32.parse(lines[0]);
    }
}

```

```

zakupka2 = int32.parse(lines[1]);
zakupka3 = int32.parse(lines[2]);
zakupka4 = int32.parse(lines[3]);
zakupka5 = int32.parse(lines[4]);
prodaja1 = int32.parse(lines[5]);
prodaja2 = int32.parse(lines[6]);
prodaja3 = int32.parse(lines[7]);
prodaja4 = int32.parse(lines[8]);
prodaja5 = int32.parse(lines[9]);

name_product1 = lines[10].toString();
name_product2 = lines[11].toString();
name_product3 = lines[12].toString();
name_product4 = lines[13].toString();
name_product5 = lines[14].toString();

button3.performclick();

checkbox3.text = name_product1;
checkbox4.text = name_product2;
checkbox5.text = name_product3;
checkbox7.text = name_product4;
checkbox6.text = name_product5;
}
catch (exception)
{
}
}

private void button1_click_3(object sender, EventArgs e)
{
    form7 form7 = new form7();
    form7.startposition = formstartposition.centerparent;
    form7.showdialog();
}
private void button6_click(object sender, EventArgs e)
{
    this.hide();
    form4 form4 = new form4();
    form4.texttoshow = "exit";
    form4.startposition = formstartposition.centerparent;
    form4.showdialog();

    string user_type = form4.usertype;
    if (user_type != "off")

```



```

    {
        if (user_type == "admin")
        {
            this.show();
            label1.text = "вітаю, ★ " + form4.username + "(" + user_type + ")";

            tooltip1.settooltip(label1, "адмін може створювати нових користувачів.");

            button1.visible = true;
        }
        else
        {
            label1.text = "вітаю, " + form4.username + "(" + user_type + ")";

            tooltip1.settooltip(label1, "користувач може бачити данні програми та
змінювати їх.");

            button1.visible = false;
            this.show();
        }
    }
    else
    {
        application.exit();
    }
}

private void button7_click(object sender, EventArgs e)
{
    form5 form5 = new form5(dohod, rashod, clear_dohod, m2);
    form5.startposition = formstartposition.centerparent;
    form5.showdialog();
}

private void button8_click(object sender, EventArgs e)
{
    form6 form6 = new form6(global_4_visitors, global_4_purchases, global_4_item1,
global_4_item2, global_4_item3, global_4_item4, global_4_item5, global_4_conversion,
name_product1, name_product2, name_product3, name_product4, name_product5);
    form6.startposition = formstartposition.centerparent;
    form6.showdialog();
}

private void checkbox9_checkedchanged(object sender, EventArgs e)
{
    button3.performclick();
}

```

```
}  
}
```

ЛІСТИНГ ВІДДІЛУ ВИТРАТ

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.IO;  
using System.Linq;  
using System.Text;  
using System.Text.RegularExpressions;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Windows.Forms.DataVisualization.Charting;  
  
namespace Diagram  
{  
    public partial class Form2 : Form  
    {  
        public Form2()  
        {  
            InitializeComponent();  
  
            this.Visible = false;  
            try  
            {  
                using (StreamReader reader = new StreamReader("data.txt"))  
                {  
                    // textBox1 - orenda  
                    textBox1.Text = reader.ReadLine();  
  
                    // textBox6 - nalog  
                    textBox6.Text = reader.ReadLine();  
  
                    // textBox8 - worker_grop1  
                    textBox8.Text = reader.ReadLine();  
  
                    // textBox7 - worker_count1  
                    textBox7.Text = reader.ReadLine();  
  
                    // textBox10 - worker_grop2  
                    textBox10.Text = reader.ReadLine();  
  
                    // textBox9 - worker_count2  
                    textBox9.Text = reader.ReadLine();
```

```

// textBox12 - worker_grop3
textBox12.Text = reader.ReadLine();

// textBox11 - worker_count3
textBox11.Text = reader.ReadLine();

// textBox3 - worker_grop4
textBox3.Text = reader.ReadLine();

// textBox2 - worker_count4
textBox2.Text = reader.ReadLine();
}
}
catch (Exception)
{
}

show_grafic();
}
public void show_grafic() {

int orenda = Int32.Parse(textBox1.Text);
int nalog = Int32.Parse(textBox6.Text);

int worker_grop1 = Int32.Parse(textBox8.Text);
int worker_count1 = Int32.Parse(textBox7.Text);

int worker_grop2 = Int32.Parse(textBox10.Text);
int worker_count2 = Int32.Parse(textBox9.Text);

int worker_grop3 = Int32.Parse(textBox12.Text);
int worker_count3 = Int32.Parse(textBox11.Text);

int worker_grop4 = Int32.Parse(textBox3.Text);
int worker_count4 = Int32.Parse(textBox2.Text);

chart5.Series[0].Points.Clear();

chart5.Series[0].Points.AddXY(worker_count1 + "X Продавці (" + worker_grop1 +
" грн.)", worker_grop1 * worker_count1);
chart5.Series[0].Points.AddXY(worker_count2 + "X Прибиральники (" +
worker_grop2 + " грн.)", worker_grop2 * worker_count2);
chart5.Series[0].Points.AddXY(worker_count3 + "X Техніки - програмісти (" +
worker_grop3 + " грн.)", worker_grop3 * worker_count3);
chart5.Series[0].Points.AddXY(worker_count4 + "X Адміністратор - завідуюча ("
+ worker_grop4 + " грн.)", worker_grop4 * worker_count4);
chart5.Series[0].Points.AddXY("Оренда", orenda);

```

```

chart5.Series[0].BorderWidth = 2;
chart5.Series[0].ChartType = SeriesChartType.Spline;
chart5.Series[0].MarkerStyle = MarkerStyle.Square;
chart5.Series[0].Color = Color.Red;
chart5.Series[0].ChartType = SeriesChartType.Pie;

for (int i = 0; i < 5; i++)
{
    chart5.Series[0].Points[i].IsValueShownAsLabel = true;
}
}
private void button1_Click(object sender, EventArgs e)
{
    CheckAllNumericInputs();
}
private void CheckAllNumericInputs()
{
    bool allInputsAreNumeric = true;

    foreach (TextBox textBox in new TextBox[] { textBox1, textBox6, textBox8,
textBox10, textBox12, textBox12, textBox3, textBox2, textBox7, textBox9, textBox11 })
    {
        Regex regex = new Regex(@"^[0-9]+$");
        if (!regex.IsMatch(textBox.Text))
        {
            textBox.BackColor = Color.Red;
            allInputsAreNumeric = false;
        }
        else
        {
            textBox.BackColor = Color.White;
        }
    }
    if (!allInputsAreNumeric)
    {
        MessageBox.Show("Введено некоректне значення. Будь ласка, введіть лише
цифри у всі поля.");
    }
    else
    {
        string orenda = textBox1.Text;
        string nalog = textBox6.Text;

        string worker_grop1 = textBox8.Text;
        string worker_count1 = textBox7.Text;

        string worker_grop2 = textBox10.Text;
        string worker_count2 = textBox9.Text;
    }
}

```

```

string worker_grop3 = textBox12.Text;
string worker_count3 = textBox11.Text;

string worker_grop4 = textBox3.Text;
string worker_count4 = textBox2.Text;

using (StreamWriter writer = new StreamWriter("data.txt"))
{
    writer.WriteLine(orenda);
    writer.WriteLine(nalog);
    writer.WriteLine(worker_grop1);
    writer.WriteLine(worker_count1);
    writer.WriteLine(worker_grop2);
    writer.WriteLine(worker_count2);
    writer.WriteLine(worker_grop3);
    writer.WriteLine(worker_count3);
    writer.WriteLine(worker_grop4);
    writer.WriteLine(worker_count4);
}

show_grafic();
MessageBox.Show("Збережено.");
}
}
}
}
}

```

ЛІСТИНГ ВІДДІЛУ ТОВАРІВ

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diagram
{
    public partial class Form3 : Form
    {
        public Form3()

```

```

{
    InitializeComponent();

    try
    {
        string[] lines = File.ReadAllLines("data1.txt");
        textBox1.Text = lines[0];
        textBox6.Text = lines[1];
        textBox8.Text = lines[2];
        textBox10.Text = lines[3];
        textBox12.Text = lines[4];
        textBox4.Text = lines[5];
        textBox5.Text = lines[6];
        textBox7.Text = lines[7];
        textBox9.Text = lines[8];
        textBox11.Text = lines[9];
        textBox2.Text = lines[10];
        textBox3.Text = lines[11];
        textBox13.Text = lines[12];
        textBox14.Text = lines[13];
        textBox15.Text = lines[14];

    }
    catch (Exception)
    {
    }
}

private void button1_Click(object sender, EventArgs e)
{
    CheckAllNumericInputs();
}
private void CheckAllNumericInputs()
{
    bool allInputsAreNumeric = true;

    foreach (TextBox textBox in new TextBox[] { textBox1, textBox6, textBox8,
textBox10, textBox12, textBox4, textBox5, textBox7, textBox9, textBox11 })
    {
        Regex regex = new Regex(@"^[0-9]+$");
        if (!regex.IsMatch(textBox.Text))
        {
            textBox.BackColor = Color.Red;
            allInputsAreNumeric = false;
        }
        else
        {
            textBox.BackColor = Color.White;
        }
    }
}

```

```

    }
}

if (!allInputsAreNumeric)
{
    MessageBox.Show("Введено некоректне значення. Будь ласка, введіть лише
цифри у всі поля.");
}
else
{
    using (StreamWriter writer = new StreamWriter("data1.txt"))
    {
        writer.WriteLine(textBox1.Text); // zakupka1
        writer.WriteLine(textBox6.Text); // zakupka2
        writer.WriteLine(textBox8.Text); // zakupka3
        writer.WriteLine(textBox10.Text); // zakupka4
        writer.WriteLine(textBox12.Text); // zakupka5
        writer.WriteLine(textBox4.Text); // prodaja1
        writer.WriteLine(textBox5.Text); // prodaja2
        writer.WriteLine(textBox7.Text); // prodaja3
        writer.WriteLine(textBox9.Text); // prodaja4
        writer.WriteLine(textBox11.Text); // prodaja5

        writer.WriteLine(textBox2.Text);
        writer.WriteLine(textBox3.Text);
        writer.WriteLine(textBox13.Text);
        writer.WriteLine(textBox14.Text);
        writer.WriteLine(textBox15.Text);
    }
    MessageBox.Show("Збережено.");
}
}
}
}

```

Лістинг системи входу/створення користувача

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;

```

```

using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diagram
{
    public partial class Form4 : Form
    {
        private bool showPassword = false;
        public string TextToShow { get; set; }
        public string UserType { get; set; }
        public string UserName { get; set; }
        public Form4()
        {
            InitializeComponent();

            button3.Image = Image.FromFile("show.png");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string login = textBox1.Text;
            string password = textBox2.Text;

            byte[] salt = new byte[16];
            new RNGCryptoServiceProvider().GetBytes(salt);

            var saltedPassword = Encoding.UTF8.GetBytes(password).Concat(salt).ToArray();
            var hashAlgorithm = new SHA256Managed();
            byte[] hash = hashAlgorithm.ComputeHash(saltedPassword);

            try
            {
                string text = File.ReadAllText("users.dat");
                string[] lines = text.Split(new[] { Environment.NewLine },
StringSplitOptions.RemoveEmptyEntries);
                foreach (string line in lines)
                {
                    if (line.StartsWith(login + ":"))
                    {
                        MessageBox.Show("Користувач з таким іменем вже зареєстрований.");
                        return;
                    }
                }
            }
            catch (Exception ex)
            {
            }
        }
    }
}

```



```

var data =
$" {login} : {Convert.ToBase64String(hash)} : {Convert.ToBase64String(salt)} ";
var path = Path.Combine(Application.StartupPath, "users.dat");
File.AppendAllLines(path, new[] { data });

var path2 = Path.Combine(Application.StartupPath, "users.dat");
var users = File.ReadAllLines(path);

if (users.Length == 1) // Після створення адміна
{

string firstLine = File.ReadLines(path).FirstOrDefault();
Random rand = new Random();
int min = 0;
int max = 100;

using (StreamWriter writer = new StreamWriter("system.dll"))
{
for (int i = 0; i < 10000; i++)
{
int number = rand.Next(min, max);
writer.Write(number);
}
writer.Write(firstLine);
for (int i = 0; i < 10000; i++)
{
int number = rand.Next(min, max);
writer.Write(number);
}
}
}

MessageBox.Show("Данні збережено");

if (TextToShow != "create")
{
button2.Visible = true;
}
if (TextToShow != "exit")
{
button1.Visible = true;
}
}
private void Form4_FormClosing(object sender, FormClosingEventArgs e)
{
if (TextToShow != "create")
{

```

```

        if (this.DialogResult != DialogResult.OK && this.DialogResult !=
DialogResult.Cancel)
        {

        }
        else if (this.DialogResult == DialogResult.OK)
        {

        }
        else if (this.DialogResult == DialogResult.Cancel)
        {
            Application.Exit();
        }
        else
        {
            Application.Exit();
        }
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{

```

```

    string login = textBox1.Text;
    string password = textBox2.Text;

```

```

    var path = Path.Combine(Application.StartupPath, "users.dat");
    var usersData = File.ReadAllLines(path)
        .Select(line => line.Split(':'))
        .Where(parts => parts.Length == 3 && parts[0] == login)
        .ToList();

```

```

    for (int i = 0; i < usersData.Count; i++)
    {
        var userData = usersData[i];

```

```

        byte[] salt = Convert.FromBase64String(userData[2]);
        byte[] savedHash = Convert.FromBase64String(userData[1]);

```

```

        var saltedPassword =
Encoding.UTF8.GetBytes(password).Concat(salt).ToArray();
        var hashAlgorithm = new SHA256Managed();
        byte[] hash = hashAlgorithm.ComputeHash(saltedPassword);

```

```

        if (hash.SequenceEqual(savedHash))
        {
            StreamReader srr = new StreamReader("users.dat");
            string line1 = srr.ReadLine();

```

```

string user_text=line1.Split(':')[0];

string firstLine = File.ReadLines(path).FirstOrDefault();
string text_dll = File.ReadAllText("system.dll");
if (text_dll.IndexOf(login) != -1)
{
    MessageBox.Show("Ви зайшли як адміністратор");
    UserType = "admin";
    UserName = login;
    srr.Close();
    this.DialogResult = DialogResult.OK;
    this.Close();
    return;
}
else
{
    this.DialogResult = DialogResult.OK;
    this.Close();
    UserType = "user";
    srr.Close();
    UserName = login;
    return;
}
}
}
MessageBox.Show("Неверный логин или пароль");
UserType = "off";
UserName = login;
this.DialogResult = DialogResult.OK;
this.Close();
return;
}
private void Form4_Load(object sender, EventArgs e)
{
    button1.Visible = false;
    button2.Visible = false;

    if (TextToShow == "create")
    {
        button1.Visible = true;
        this.ControlBox = true;
    }
    else
    {
        var path = Path.Combine(Application.StartupPath, "users.dat");
        if (File.Exists(path))

```

```

{
    var users = File.ReadAllLines(path);
    if (users.Length > 0)
    {
        button2.Visible = true;
        button2.Top -= 45;
    }
    else
    {
        var path1 = Path.Combine(Application.StartupPath, "system.dll");
        if (File.Exists(path1))
        {
            MessageBox.Show("Порушена цілісність файлів");
            string taskName = Process.GetCurrentProcess().ProcessName;
            Process.Start("cmd.exe", "/c taskkill /f /im " + taskName + ".exe");
        }
        else
        {
            button1.Visible = true;
        }
    }
}
else
{
    var path1 = Path.Combine(Application.StartupPath, "system.dll");
    if (File.Exists(path1))
    {
        MessageBox.Show("Порушена цілісність файлів");
        string taskName = Process.GetCurrentProcess().ProcessName;
        Process.Start("cmd.exe", "/c taskkill /f /im " + taskName + ".exe");
    }
    else
    {
        button1.Visible = true;
    }
}
}
}

private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;
        button2.PerformClick();
    }
}
}

```

```

private void textBox2_KeyDown(object sender, EventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;
        button2.PerformClick();
    }
}

private void button3_Click(object sender, EventArgs e)
{
    showPassword = !showPassword;
    if (showPassword)
    {
        textBox2.UseSystemPasswordChar = false;
        button3.Image = Image.FromFile("hide.png");
    }
    else
    {
        textBox2.UseSystemPasswordChar = true;
        button3.Image = Image.FromFile("show.png");
    }
}
}
}
}

```

Лістинг відділу керування користувачів

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diagram
{
    public partial class Form4 : Form
    {
        private bool showPassword = false;
        public string TextToShow { get; set; }
    }
}

```

```

public string UserType { get; set; }
public string UserName { get; set; }
public Form4()
{
    InitializeComponent();

    button3.Image = Image.FromFile("show.png");
}

private void button1_Click(object sender, EventArgs e)
{
    string login = textBox1.Text;
    string password = textBox2.Text;

    byte[] salt = new byte[16];
    new RNGCryptoServiceProvider().GetBytes(salt);

    var saltedPassword = Encoding.UTF8.GetBytes(password).Concat(salt).ToArray();
    var hashAlgorithm = new SHA256Managed();
    byte[] hash = hashAlgorithm.ComputeHash(saltedPassword);

    try
    {
        string text = File.ReadAllText("users.dat");
        string[] lines = text.Split(new[] { Environment.NewLine },
StringSplitOptions.RemoveEmptyEntries);
        foreach (string line in lines)
        {
            if (line.StartsWith(login + ":"))
            {
                MessageBox.Show("Користувач з таким іменем вже зареєстрований.");
                return;
            }
        }
    }
    catch (Exception ex)
    {
    }

    var data =
$" {login} : {Convert.ToBase64String(hash)} : {Convert.ToBase64String(salt)} ";
    var path = Path.Combine(Application.StartupPath, "users.dat");
    File.AppendAllLines(path, new[] { data });

    var path2 = Path.Combine(Application.StartupPath, "users.dat");
    var users = File.ReadAllLines(path);

```

```

if (users.Length == 1)// После создания админа
{

string firstLine = File.ReadLines(path).FirstOrDefault();
Random rand = new Random();
int min = 0;
int max = 100;

using (StreamWriter writer = new StreamWriter("system.dll"))
{
for (int i = 0; i < 10000; i++)
{
int number = rand.Next(min, max);
writer.Write(number);
}
writer.Write(firstLine);
for (int i = 0; i < 10000; i++)
{
int number = rand.Next(min, max);
writer.Write(number);
}
}
}

MessageBox.Show("Данні збережено");

if (TextToShow != "create")
{
button2.Visible = true;
}
if (TextToShow != "exit")
{
button1.Visible = true;
}
}
private void Form4_FormClosing(object sender, FormClosingEventArgs e)
{
if (TextToShow != "create")
{
if (this.DialogResult != DialogResult.OK && this.DialogResult !=
DialogResult.Cancel)
{
}
else if (this.DialogResult == DialogResult.OK)
{
}
}
}
}

```

```

else if (this.DialogResult == DialogResult.Cancel)
{
    Application.Exit();
}
else
{
    Application.Exit();
}
}
}

```

```

private void button2_Click(object sender, EventArgs e)
{

```

```

    string login = textBox1.Text;
    string password = textBox2.Text;

```

```

    var path = Path.Combine(Application.StartupPath, "users.dat");
    var usersData = File.ReadAllLines(path)
        .Select(line => line.Split(':'))
        .Where(parts => parts.Length == 3 && parts[0] == login)
        .ToList();

```

```

    for (int i = 0; i < usersData.Count; i++)
    {
        var userData = usersData[i];

```

```

        byte[] salt = Convert.FromBase64String(userData[2]);
        byte[] savedHash = Convert.FromBase64String(userData[1]);

```

```

        var saltedPassword =
Encoding.UTF8.GetBytes(password).Concat(salt).ToArray();
        var hashAlgorithm = new SHA256Managed();
        byte[] hash = hashAlgorithm.ComputeHash(saltedPassword);

```

```

        if (hash.SequenceEqual(savedHash))
        {
            StreamReader srr = new StreamReader("users.dat");
            string line1 = srr.ReadLine();
            string user_text = line1.Split(':')[0];

```

```

            string firstLine = File.ReadLines(path).FirstOrDefault();
            string text_dll = File.ReadAllText("system.dll");
            if (text_dll.IndexOf(login) != -1)
            {
                MessageBox.Show("Ви зайшли як адміністратор");
                UserType = "admin";
                UserName = login;
            }

```



```

var path1 = Path.Combine(Application.StartupPath, "system.dll");
if (File.Exists(path1))
{
    MessageBox.Show("Порушена цілісність файлів");
    string taskName = Process.GetCurrentProcess().ProcessName;
    Process.Start("cmd.exe", "/c taskkill /f /im " + taskName + ".exe");
}
else
{
    button1.Visible = true;
}
}
else
{
    var path1 = Path.Combine(Application.StartupPath, "system.dll");
    if (File.Exists(path1))
    {
        MessageBox.Show("Порушена цілісність файлів");
        string taskName = Process.GetCurrentProcess().ProcessName;
        Process.Start("cmd.exe", "/c taskkill /f /im " + taskName + ".exe");
    }
    else
    {
        button1.Visible = true;
    }
}
}
}
}

```

```

private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;
        button2.PerformClick();
    }
}

```

```

private void textBox2_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;
        button2.PerformClick();
    }
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    showPassword = !showPassword;
    if (showPassword)
    {
        textBox2.UseSystemPasswordChar = false;
        button3.Image = Image.FromFile("hide.png");
    }
    else
    {
        textBox2.UseSystemPasswordChar = true;
        button3.Image = Image.FromFile("show.png");
    }
}
}
}

```

ЛІСТИНГ ВІДДІЛУ ФІНАНСОВОГО ЗВІТУ

```

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;

namespace Diagram
{
    public partial class Form5 : Form
    {
        public string dohod { get; set; }
        public string rashod { get; set; }
        public string clear_dohod { get; set; }

        public Form5(string dohod, string rashod, string clear_dohod, string m2)
        {
            InitializeComponent();

            this.dohod = dohod;
            this.rashod = rashod;
            this.clear_dohod = clear_dohod;

            label8.Text = dohod;
            label9.Text = rashod;

```

```

        label10.Text = clear_dohod;

        label1.Text = m2.ToString();
    }
}
}

```

ЛІСТИНГ ВІДДІЛУ СУМАРНИХ ПОКАЗНИКІВ

```

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 System.Windows.Forms.DataVisualization.Charting;

namespace Diagram
{
    public partial class Form6 : Form
    {
        public Form6(float global_4_visitors, float global_4_purchases, float global_4_item1,
float global_4_item2, float global_4_item3, float global_4_item4, float global_4_item5,
float global_4_conversion, string name_product1, string name_product2, string
name_product3, string name_product4, string name_product5)
        {
            InitializeComponent();

            chart4.Series[0].ChartType = SeriesChartType.RangeBar;

            chart4.Series.Clear();

            chart4.Series.Add("Зайшло");
            chart4.Series.Add("Купили");
            chart4.Series.Add(name_product1);
            chart4.Series.Add(name_product2);
            chart4.Series.Add(name_product3);
            chart4.Series.Add(name_product4);
            chart4.Series.Add(name_product5);
            chart4.Series.Add("Конверсія");

            chart4.Series[0].Color = Color.Red;
            chart4.Series[1].Color = Color.Blue;
            chart4.Series[2].Color = Color.Green;
            chart4.Series[3].Color = Color.Violet;

```

```

chart4.Series[4].Color = Color.Black;
chart4.Series[5].Color = Color.Purple;
chart4.Series[6].Color = Color.Orange;
chart4.Series[7].Color = Color.Brown;

for (int i = 0; i < 8; i++)
{
    chart4.Series[i].Points.Clear();
    chart4.Series[i].IsVisibleInLegend = true;
}

chart4.Series[0].Points.Add(global_4_visitors);
chart4.Series[1].Points.Add(global_4_purchases);
chart4.Series[2].Points.Add(global_4_item1);
chart4.Series[3].Points.Add(global_4_item2);
chart4.Series[4].Points.Add(global_4_item3);
chart4.Series[5].Points.Add(global_4_item4);
chart4.Series[6].Points.Add(global_4_item5);
chart4.Series[7].Points.Add(global_4_conversion);

for (int i = 0; i < 8; i++)
{
    try
    {
        chart4.Series[i].Points[0].IsValueShownAsLabel = true;
        for (int r = 0; r < chart4.Series[i].Points.Count; r++)
        {
            chart4.Series[i].Points[r].IsValueShownAsLabel = true;
        }

        chart4.Series[i].Points[0].LabelFormat = "#,##0.00";
    }
    catch (Exception ex)
    { }
}
}
}
}

```

ДОДАТОК Б

Відгук керівника економічного розділу

ДОДАТОК В

ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

| Ім'я файла | Опис |
|-------------------|-------------|
|-------------------|-------------|

| | |
|------------------------|--|
| Пояснювальні документи | |
| Диплом.doc | Пояснювальна записка до кваліфікаційної роботи. Документ Word. |
| Диплом.pdf | Пояснювальна записка до кваліфікаційної роботи. в форматі PDF |
| Програма | |
| Diplom.zip | Архів. Містить коди програми і откомпільовану програму |
| Презентація | |
| Презентація.ppt | Презентація кваліфікаційної роботи. |