

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ центр післядипломної освіти \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Інформаційна система обробки замовлень дизайн-студії поліграфічного підприємства  
(тема)

Виконав:

студент 4 курсу, групи ПЗПп-22-1

\_\_\_\_\_ Соколов А. Ю.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва спеціальності)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Освітня програма Програмна інженерія

(повна назва освітньої програми)

Керівник к.т.н., доцент кафедри ПІ Каук В. І.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_

(підпис)

\_\_\_\_\_ З.В.Дудар

(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ центр післядипломної освіти \_\_\_\_\_  
 Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_  
 Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_  
 Спеціальність \_\_\_\_\_ 121 – Інженерія програмного забезпечення \_\_\_\_\_  
 Тип програми \_\_\_\_\_ Освітньо-професійна \_\_\_\_\_  
 Освітня програма \_\_\_\_\_ Програмна Інженерія \_\_\_\_\_  
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
 (підпис)  
 «\_\_\_\_» \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Соколову Андрію Юрійовичу \_\_\_\_\_  
 (прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система обробки замовлень дизайн-студії поліграфічного підприємства

Затверджена наказом по університету від 06.05.2024р. № 401 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14.06.2024

3. Вихідні дані до роботи.

Операційна система: ОС Windows.

Мова програмування: C#.

Система управління базами даних: Microsoft SQL Server 2022.

Бібліотеки та фреймворки: Entity Framework.

4. Перелік питань, що потрібно опрацювати в роботі

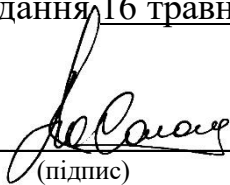
Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	20.05.2024	<i>виконано</i>
2	Створення специфікації ПЗ	22.05.2024	<i>виконано</i>
3	Проектування ПЗ	24.05.2024	<i>виконано</i>
4	Розробка ПЗ	28.05.2024	<i>виконано</i>
5	Тестування ПЗ	30.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	05.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	06.06.2024	<i>виконано</i>
8	Попередній захист	20.07.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	15.07.2024	<i>виконано</i>
10	Здача роботи у електронний архів	20.07.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	22.07.2024	<i>виконано</i>

Дата видачі завдання 16 травня 2024р.

Студент (ка) \_\_\_\_\_

  
(підпис)

Соколов А. Ю.

Керівник роботи \_\_\_\_\_

(підпис)

к.т.н., доцент кафедри ПІ Каук В. І.

(посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра, 59 стор., 7 рис., 5 табл., 11 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМНИЙ ЗАСТОСУНОК, C#, SQL, РЕЛЯЦІЙНІ БАЗИ ДАНИХ

Об'єкт дослідження: організаційні та бізнес процеси поліграфічного підприємства, що потребують автоматизації.

Мета розробки: Створення програмної системи, що забезпечить ефективну обробку замовлень у дизайн-студії поліграфічного підприємства.

Метод рішення: Використання технологій .NET для розробки програмного забезпечення для настільних комп'ютерів. В якості системи управління базами даних обрано Microsoft SQL Server 2022.

Результати розробки: Розроблена програмна система, яка дозволяє додавати, змінювати та видаляти інформацію в базі даних, а також формувати необхідні документи внутрішнього формату підприємства.

INFORMATION SYSTEM OF ORDER PROCESSING, DESIGN STUDIO, PRINTING COMPANY, C#, WINDOWS FORM, SQL

Object of research: organizational and business processes of a printing enterprise that require automation.

Development goal: Creation of a software system that will ensure efficient processing of orders in the design studio of a printing company.

Solution Method: Use Windows Visual Studio to develop desktop software. Microsoft SQL Server 2022 was chosen as the database management system.

Development results: A software system has been developed that allows you to add, change, and delete information in the database, as well as create the necessary documents of the company's internal format.

Я, Соколов Андрій Юрійович, студент гр. ПЗПп-22-1, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Інформаційна система обробки замовлень дизайн-студії поліграфічного підприємства», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## Зміст

Вступ.....	7
1 Аналіз предметної галузі .....	10
1.1 Аналіз предметної галузі .....	10
1.2 Виявлення та вирішення проблем .....	13
1.3 Постановка задачі.....	13
2 Формування вимог до програмної системи .....	16
3 Архітектура та проектування програмного забезпечення .....	21
3.1 Архітектурні підходи .....	21
3.1.1 Рівень презентації (ui/ux) .....	23
3.1.2 Рівень бізнес-логіки .....	23
3.1.3 Рівень доступу до даних .....	23
3.1.4 рівень зберігання даних .....	24
3.2 проектування архітектури.....	24
3.2.1 Діаграма класів .....	24
3.2.2 Діаграма компонентів .....	25
3.2.3 Діаграма розгортання.....	25
3.3 Проектування структури зберігання даних .....	25
3.4 Приклади найцікавіших алгоритмів та методів.....	25
3.5 Створення ui/ux .....	27
4 Опис прийнятих програмних рішень .....	29
4.1 Створення програмного додатка .....	29
5 Тестування програмного забезпечення.....	35
5.1 Тестування застосунку.....	39
6 Впровадження програмного забезпечення .....	44
6.1 Визначення плану впровадження .....	44
Висновки.....	48
Перелік джерел посилання .....	50
Додаток А .....	52
Додаток Б.....	53
Додаток В .....	59

## ВСТУП

Актуальність та важливість цієї роботи визначаються потребою поліграфічних підприємств у вдосконаленні процесів управління замовленнями та підвищенні ефективності виробництва. Розробка додатку "Програмна система обробки замовлень дизайн-студії поліграфічного підприємства" є відповіддю на потребу в модернізації та автоматизації бізнес-процесів в цій галузі.

Застосування реляційних баз даних сприяє збереженню та організації великої кількості інформації про замовлення, клієнтів та інші аспекти діяльності студії. Логіка об'єктно-орієнтованого програмування дозволяє створити модульну та гнучку систему, яка легко масштабується та розвивається з ростом бізнесу.

Такий підхід до розробки програмного забезпечення дозволить оптимізувати робочі процеси, підвищити продуктивність персоналу та забезпечити високий рівень обслуговування клієнтів. У результаті, ця інформаційна система стане ключовим інструментом для поліпшення конкурентоспроможності поліграфічного підприємства на ринку.

У ході створення додатку були використані середовище розробки Microsoft Visual Studio 2022, платформа ASP.Net, мова програмування C#, інтерфейс програмування додатків Windows Forms та система управління базами даних Microsoft SQL Server 2022[1].

У сучасному світі, де конкуренція на ринку постійно зростає, важливо мати ефективні та інноваційні інструменти для підтримки бізнесу. Для дизайн-студій поліграфічних підприємств це особливо актуально, оскільки вони працюють у галузі, де важлива креативність, точність та швидкість реакції на потреби клієнтів. У цьому контексті розробка програмного забезпечення, яке підтримує обробку замовлень та управління даними, має велике значення, оскільки надає можливість

зберігати, аналізувати та оптимізувати великі обсяги інформації, що сприяє удосконаленню виробничих процесів та підвищенню ефективності роботи студії. Таке програмне забезпечення дозволяє автоматизувати багато рутинних операцій, зменшуючи час, необхідний для їх виконання, та мінімізуючи ризик помилок. У результаті, компанія може швидше реагувати на зміни в запитах клієнтів, забезпечуючи високу якість обслуговування та задоволення їх потреб.

Успішне використання інформаційних технологій для автоматизації робочих процесів в сучасному бізнесі є ключовим фактором у досягненні успіху та конкурентоспроможності. Використання спеціалізованого програмного забезпечення, яке оптимізує обробку замовлень та забезпечує збереження даних про технічні вимоги, дизайн, клієнтів та інші аспекти роботи, дозволяє підприємствам підвищити продуктивність, зменшити час, потрібний для виконання завдань, та підвищити рівень задоволеності клієнтів шляхом забезпечення високої якості обслуговування.

Ручне ведення даних не лише годино - та працезатратне, але й небезпечне через високий ризик помилок та неузгодженостей у інформації. Навіть найдрібніша помилка у веденні даних може мати серйозні наслідки для діяльності підприємства, зокрема, призвести до невірних рішень, втрати клієнтів або навіть збитків. Використання системи управління базами даних дозволяє ефективно вирішити цю проблему, забезпечуючи точність та надійність даних. Вона автоматизує процеси ведення та збереження інформації, уникнення помилок та дублювання даних, що робить її незамінною для діяльності дизайн-студій поліграфічного підприємства.

Розробка програмного забезпечення, яке включає в себе функціонал пошуку та фільтрації інформації, а також надає можливість отримання детальної статистики про виконані роботи, має великий потенціал у підвищенні продуктивності та зниженні витрат на робочу силу. Завдяки автоматизації процесів, які раніше вимагали значних людських зусиль,

працівники можуть ефективніше виконувати свої обов'язки, витрачаючи менше часу на монотонні та рутинні завдання. Крім того, можливість автоматичної розсилки повідомлень на електронну пошту клієнтів та співробітників дозволяє підприємству швидко та ефективно спілкуватися зі своїми контактами, що сприяє покращенню комунікацій та підвищенню задоволеності клієнтів. Такі інструменти стають невід'ємною частиною сучасного бізнесу, дозволяючи підприємствам ефективно управляти своїми ресурсами та досягати більшого успіху на ринку.

У сучасному світі, де швидкість, точність та ефективність є ключовими факторами успіху, важливо мати належні інструменти для підтримки робочих процесів. Для дизайн-студій поліграфічних підприємств це особливо актуально, оскільки їхня робота напряму пов'язана з творчістю, високими вимогами до якості та швидкими термінами виконання.

Впровадження сучасних технологій у їхню роботу дозволяє значно підвищити продуктивність та якість виконаної роботи. Інтеграція інструментів розробки програмного забезпечення, платформ програмування та систем управління базами даних стає важливим етапом у вдосконаленні робочих процесів.

Використання спеціалізованих програмних рішень дозволяє автоматизувати багато рутинних операцій та забезпечити точність у веденні та обробці даних. Це дозволяє зосередитися на творчих аспектах роботи та зменшує ймовірність помилок..

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Аналіз предметної галузі

Дизайн-студія поліграфічного підприємства потребує ефективного інструменту для обробки та керування замовленнями. Система допомагає у керуванні технічними завданнями, дизайном, інформацією про клієнтів та іншими аспектами замовлень. Система має наступні функціональні вимоги:

- реєстрація та зберігання інформації про замовлення, включаючи технічні вимоги, дизайнерські концепції та контактні дані клієнтів;
- можливість класифікації та категоризації замовлень для швидкого пошуку та фільтрації;
- автоматизована генерація звітів про стан замовлень, продуктивність та інші важливі метрики.

До нефункціональних вимог можна віднести:

- висока надійність та безпека даних, включаючи захист від несанкціонованого доступу та резервне копіювання;
- інтуїтивний та зручний інтерфейс користувача для простого та швидкого доступу до інформації;
- швидкодія системи та відмінна продуктивність навіть при великому обсязі даних.

Умови впровадження:

- планована дата запуску системи;
- наявність необхідних ресурсів для розгортання та підтримки системи;
- проведення навчання персоналу щодо використання нової системи.

Програма дозволяє оформлювати списання оприбуткування залишків та інвентаризацію. Програма дозволяє налаштовувати інтерфейс та рівні доступу до даних в залежності від ролі користувача.

Наразі існує багато застосунків для ведення обліку на підприємстві. Серед використовуваних можна визначити «1С: Підприємство» [2].

На рисунку 1 ви можете побачити приклад інтерфейсу програми.

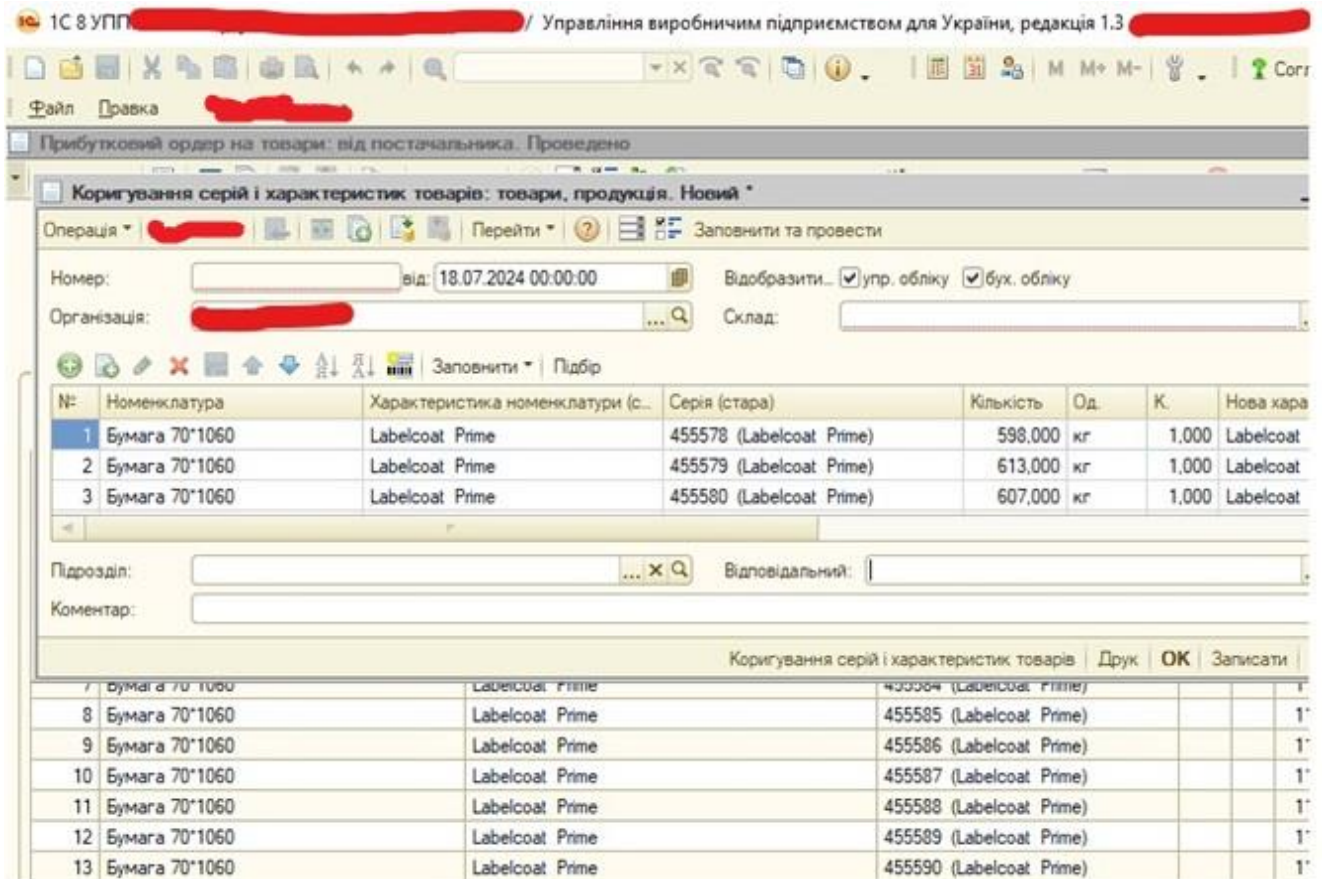


Рисунок 1 – Приклад інтерфейсу (Виконано самостійно)

Нижче, декілька інших аналогів програмної системи, які можуть бути корисними для обробки замовлень у дизайн-студії поліграфічного підприємства:

1. SAP Business One - це програмна система управління підприємством, яка має модулі для управління клієнтами, фінансами, продажами, складом та виробництвом. Вона може бути адаптована для вирішення потреб дизайн-студії [3].

2. Microsoft Dynamics 365 - це інтегрована платформа управління відносинами з клієнтами (CRM) та управління ресурсами підприємства (ERP). Вона надає широкі можливості для обробки замовлень та ведення ділової діяльності.

3. Odoo - це відкрита програмна система управління бізнесом, яка має модулі для різних аспектів діяльності підприємства, включаючи CRM, управління проектами, склад, фінанси та інше. Вона може бути налаштована для потреб дизайн-студії.

4. NetSuite - це хмарна платформа управління підприємством, яка має модулі для фінансів, CRM, управління продажами, виробництва та інших аспектів бізнесу. Вона може бути використана для обробки замовлень у дизайн-студії.

5. Bitrix24 - це інтегрована платформа для управління відносинами з клієнтами, співпраці та управління проектами. Вона містить інструменти для обробки замовлень та ведення проектів у дизайн-студії [4].

Ці системи мають свої унікальні особливості та можливості, і вони можуть бути адаптовані для вирішення конкретних потреб вашої дизайн-студії поліграфічного підприємства.

Найбільш відповідає тематиці даної практики - «1С: Підприємство».

Програмна система «1С: Підприємство» є однією з найпоширеніших інтегрованих платформ для управління бізнесом в Україні та країнах СНД. Її популярність зумовлена широким функціоналом, що охоплює майже всі аспекти діяльності підприємства. Однією з головних переваг "1С: Підприємство" є гнучкість налаштування під індивідуальні потреби кожного підприємства. Система дозволяє створювати та налаштовувати різні конфігурації з урахуванням специфіки галузей, що відповідають потребам користувачів [2].

До інших переваг програми варто віднести розгалужену систему модулів, що охоплюють всі основні аспекти управління підприємством, від фінансів та обліку до управління складом, замовленнями та виробництвом. Крім того, «1С: Підприємство» має велику спільноту користувачів та розробників, що забезпечує широкий доступ до консультантів та підтримки.

Проте, важливо відзначити деякі недоліки програми. Одним з них є складність системи конфігурації та налаштування. Це може вимагати значних зусиль та часу для вивчення та впровадження системи на підприємстві. Крім того,

при оновленні програмного забезпечення можуть виникати конфлікти та невідповідності з попередніми версіями, що може потребувати додаткових зусиль для вирішення.

Усупереч цим недолікам, програмна система «1С: Підприємство» залишається одним з найпопулярніших та найбільш ефективних рішень для управління підприємством в умовах сучасного бізнесу. Її гнучкість, широкий функціонал та підтримка з боку спільноти користувачів роблять її привабливим вибором для багатьох підприємств.

За допомогою даного застосунку менеджери підприємств можуть контролювати надходження товарів на склад, переміщення у рамках складу, вибуття товару та сплату за товар.

## 1.2 Виявлення та вирішення проблем

Існує досить багато поліграфічних підприємств, різної направленості, які мають у своєму складі окремий відділ, що називається дизайн студія. Це може бути як внутрішній підрозділ, включений до фінансово – економічної структури підприємства, так і окрема компанія, яка бере на себе даний вид робіт. Наявність великої кількості замовників, кожен із яких має ряд лінійок, найменувань готової продукції призводить до необхідності наявності великого місця зберігання різноманітності файлів. У разі обслуговування дизайн студією крім одного підприємства, так само інших підприємств, це може призвести до втрат часу на пошук та сортування та збереження робіт у певному алгоритмі. Наприклад, декілька підприємств, які мають свої студії для обробки замовлень для подальшого виробництва поліграфічної продукції, в яких до роботи потрапляє декілька тисяч технічних замовлень, які можуть містити до 10 видів дизайнів. Саме тому, це збільшення обсягу найменувань, та можливість зазирнути всередину процесів ведення управлінського обліку здалась мені актуальною та цікавою. Визначена проблема, може бути вирішена за допомогою обраної теми.

## 1.3 Постановка задачі

Мета роботи полягає у розробці програмної системи, призначеної для оптимізації процесів обробки замовлень в дизайн-студії поліграфічного підприємства. Завдання включають в себе:

- розробку зручного та інтуїтивно зрозумілого інтерфейсу, що дозволить користувачам легко взаємодіяти з системою;
- реалізацію функціоналу для реєстрації та зберігання інформації про замовлення, включаючи технічні вимоги, дизайнерські концепції та контактні дані клієнтів;
- створення можливості класифікації та категоризації замовлень для швидкого пошуку та фільтрації;
- розробку автоматизованих засобів генерації звітів про стан замовлень, продуктивність та інші важливі метрики;
- забезпечення високої надійності та безпеки даних, включаючи захист від несанкціонованого доступу та резервне копіювання;
- інтеграцію з існуючими системами управління, що дозволить ефективно обмінюватися даними та спільно використовувати ресурси.

Ці завдання спрямовані на створення інформаційної системи, яка допоможе поліпшити продуктивність та ефективність роботи дизайн-студії поліграфічного підприємства, спростить процеси обробки замовлень та підвищить рівень задоволеності клієнтів.

Для досягнення цих цілей було використано ряд сучасних технологій та методологій розробки програмного забезпечення. Зокрема використана мова програмування C# у поєднанні з фреймворком .NET для реалізації бізнес-логіки та розробки інтерфейсу користувача. В якості системи управління базами даних буде використана Microsoft SQL Server, що забезпечить швидкий та надійний доступ до даних [5]. Також використовуються UML-діаграм для моделювання архітектури системи та визначення взаємозв'язків між її складовими частинами. Крім того, для забезпечення якості програмного забезпечення проведено ретельне

тестування, включаючи модульне та інтеграційне тестування, а також тестування в реальних умовах експлуатації.

## 2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Інформація щодо формування вимог до програмної системи.

Можна розділити вимоги на функціональні та не функціональні, а також особливі.

Тобто функціональні вимоги, можна описати як опис функціональності програмного забезпечення, включаючи усі операції та функції, які воно повинно виконувати.

Як приклад, визначення можливостей системи щодо обробки замовлень, включаючи реєстрацію, зберігання, пошук та аналітику.

Реєстрація замовлень. Система повинна надавати можливість користувачам реєструвати нові замовлення в системі шляхом заповнення відповідних форм.

Зберігання інформації про замовлення. Після реєстрації замовлення, система повинна зберігати всю необхідну інформацію про нього, включаючи технічні деталі, дату та час реєстрації, статус тощо.

Пошук та фільтрація замовлень. Користувачам має бути надана можливість шукати замовлення за різними критеріями, такими як дата, статус, тип тощо, а також застосовувати фільтри для точного визначення потрібного набору замовлень.

Генерація звітів. Система повинна мати можливість генерувати різноманітні звіти про стан замовлень, продуктивність, фінансові показники тощо, які допоможуть користувачам отримати необхідну інформацію для управління бізнесом.

Керування користувачами. Адміністраторам системи має бути надана можливість керувати правами доступу користувачів, створювати нові облікові записи, блокувати або видаляти користувачів за необхідності.

Ці функціональні вимоги визначають основні можливості програмної системи та її функціонал для задоволення потреб користувачів та досягнення цілей практики.

До нефункціональних вимог можна віднести визначення характеристик системи, таких як продуктивність, безпека, надійність, доступність та інші.

Отже продуктивність. Наша система повинна забезпечувати високу швидкість обробки та відповіді на запити користувачів навіть при великому обсязі даних та навантаженні на сервер.

При цьому, вона має забезпечити високого рівня безпеку даних, шляхом використання шифрування, аутентифікації користувачів та контролю доступу до конфіденційної інформації.

Система повинна бути надійною, стійкою до відмов та забезпечувати безперебійну роботу навіть при виникненні непередбачуваних ситуацій, таких як відключення електропостачання або помилки в програмному забезпеченні.

Також вона повинна мати доступність. Забезпечення доступності системи для користувачів у будь-який час та з будь-якого місця, включаючи можливість працювати з системою в режимі онлайн та офлайн.

Що до інтерфейсу користувача нашого програмного забезпечення. Його розробка має бути зручна та приваблива користувачеві, що сприятиме зручному та ефективному використанню системи.

Наше програмне забезпечення має сумісності з різними операційними системами, браузерами та пристроями, що дозволить користувачам працювати з системою з будь-якого пристрою за їхніми вподобаннями.

Опис вимог щодо інтерфейсу користувача, включаючи зручність використання, естетичний дизайн та сумісність з різними пристроями та розширеннями.

До особливих вимог можна врахувати специфічні потреби користувачів, таких як можливість спільної роботи декількох користувачів над одним замовленням.

Система повинна бути здатна інтегруватися з існуючими системами управління, що використовуються на поліграфічному підприємстві, такими як системи управління складом, обліку замовлень або фінансові програми.

Система має можливість забезпечити масштабування системи, для врахування зростаючих потреб бізнесу та обсягів даних, що дозволить забезпечити ефективне функціонування системи у майбутньому.

Розробка програмного забезпечення, що підтримує різні платформи, такі як робочі станції з ОС Windows та macOS, а також мобільні пристрої з ОС Android та iOS, для забезпечення доступу до системи з будь-якого пристрою.

Система має можливість впровадження автоматичного резервного копіювання даних, що забезпечить збереження інформації та уникне втрати даних у випадку аварійних ситуацій або випадкового видалення.

Розробка програмного забезпечення з використанням багаторівневої архітектури, що дозволить розділити логіку бізнес-процесів, інтерфейс користувача та роботу з базою даних для забезпечення кращої модульності та масштабованості системи.

Ці особливі вимоги враховують специфіку діяльності поліграфічного підприємства та забезпечать високий рівень функціональності, ефективності та надійності програмного забезпечення.

Наше програмне рішення складається із двох компонентів.

По-перше – це сама база даних. База даних розроблена СУБД база даних з використанням SQL мови програмування.

Переваги SQL баз даних.

SQL (Structured Query Language) є стандартизованою мовою для роботи з реляційними базами даних. Це означає, що розробка та управління базами даних, написання запитів і зміни в базі даних можуть бути здійснені незалежно від конкретної системи управління базами даних (СУБД).

SQL бази даних володіють високою продуктивністю та швидкістю обробки запитів. Це досягається завдяки оптимізації запитів, використанню індексів та інших методів оптимізації для прискорення вибірки, сортування та з'єднання даних.

SQL дозволяє виконувати складні запити до бази даних, включаючи з'єднання, групування, фільтрацію, сортування та агрегацію даних. Це робить SQL потужним інструментом для аналізу та вибірки необхідних даних.

SQL бази даних надають високий рівень безпеки для збереження та захисту конфіденційних даних. Вони забезпечують можливість налаштування доступу до даних для користувачів і ролей, шифрування даних та контроль цілісності бази даних.

SQL бази даних можуть легко масштабуватися, щоб відповідати зростаючим потребам організації. Вони підтримують можливість додавання нових таблиць, індексів та реплікації для підтримки більшої кількості даних та високої завантаженості.

SQL бази даних мають вбудовані механізми резервного копіювання та відновлення даних. Це дозволяє забезпечити безпеку даних та можливість відновити базу даних у випадку виникнення помилок або втрати даних.

Для оптимальної роботи інформаційної системи, вона буде інтегрована з сервером баз даних Microsoft SQL Server. SQL Server відомий своєю надійністю, швидкістю та широким набором функцій, що робить його популярним в області розробки програмного забезпечення. Багато розробників вже мають досвід роботи з SQL Server, що значно спрощує процес розробки та підтримки системи. Крім того, інтерфейс та інструменти адміністрування SQL Server також є дружніми до користувача, що сприяє швидкому вивченню та ефективному використанню цієї технології.

Другий компонент нашої системи це графічний інтерфейс.

Розробляється графічний інтерфейс користувача на основі Windows Forms, який буде зручним та інтуїтивно зрозумілим.

Додаток на основі Windows Forms (Windows Forms Application) має кілька переваг:

- легкість розробки: Windows Forms надає розробникам простий та зрозумілий інструментарій для створення графічного інтерфейсу користувача. За допомогою візуального дизайнера можна швидко розміщувати елементи керування (такі як кнопки, текстові поля, списки тощо) на формі та налаштовувати їх властивості;
- широкий набір елементів керування: Windows Forms надає розширений набір вбудованих елементів керування, які можна використовувати для побудови

інтерактивного інтерфейсу користувача. Це дозволяє створювати різноманітні форми, вікна та діалогові вікна, що задовольняють потреби користувачів;

- широке розповсюдження: Windows Forms є добре відомим та поширеним інструментом для розробки Windows-додатків. Це означає, що існує велика кількість ресурсів, документації, форумів та готових рішень, які допоможуть розробникам у процесі створення додатку;

- інтеграція з .NET Framework: Windows Forms базується на .NET Framework, що надає широкі можливості для розробки додатків. Розробники можуть використовувати багатофункціональну бібліотеку класів .NET, доступ до бази даних, роботу з мережею та інші функціональні можливості, що робить розробку додатків більш гнучкою та потужною;

- підтримка подій та обробників подій: Windows Forms надає механізм подій та обробників подій, що дозволяє реагувати на взаємодію користувача з додатком. Це дозволяє створювати відгук на дії користувача, такі як натискання кнопок, введення даних тощо, і виконувати відповідні дії;

- кросплатформеність: Хоча Windows Forms розроблено для платформи Windows, існують також способи виконання додатків Windows Forms на інших платформах за допомогою технологій, таких як Mono. Це може бути важливим фактором для розробки додатків, які потребують кросплатформеності.

Ці переваги роблять Windows Forms популярним вибором для розробки додатків з графічним інтерфейсом на платформі Windows. Однак, слід зазначити, що за останніми роками увага розробників все більше переходить до більш сучасних технологій, таких як Windows Presentation Foundation (WPF) та Universal Windows Platform (UWP).

Використовуємо візуальні елементи керування, такі як кнопки, поля введення та таблиці, для забезпечення зручності взаємодії з програмою.

### 3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Основні функціональні модулі у Windows Forms додатку вже розроблені, та включають модулі для авторизації користувачів, керування дизайнами, звітності тощо.

Модуль авторизації може передавати інформацію про користувача модулю керування дизайнами для доступу до відповідних функцій. Перелік функцій включає в себе:

- додавання інформації по користувачам;
- розподіл їх обов'язків;
- розподіл їх та навиків.

Модуль керування дизайнами виконує операції створення, редагування та видалення дизайнів, а також здійснювати зв'язок з базою даних для збереження і отримання даних.

Модуль взаємодії з іншими відділами. Так як програмне забезпечення що розроблено і буде використовуватимуся в окремому відділі підприємства, цей модуль буде включати звіти, формування технічних специфікацій для подальшого використання в технологічному циклі підприємства.

Архітектура та проєктування програмного забезпечення є ключовими етапами в розробці інформаційної системи, оскільки визначають структуру системи, взаємодію між її компонентами та способи їх реалізації. У даному розділі розглянемо основні аспекти архітектури та проєктування програмного забезпечення для системи обробки замовлень дизайн-студії поліграфічного підприємства.

#### 3.1 Архітектурні підходи

При розробці архітектури нашої системи ми використовували багаторівневу архітектуру, яка складається з кількох основних рівнів:

- рівень презентації (UI/UX);

- рівень бізнес-логіки;
- рівень доступу до даних;
- рівень зберігання даних.

Use-case діаграма застосунку представлена на рисунку 2.

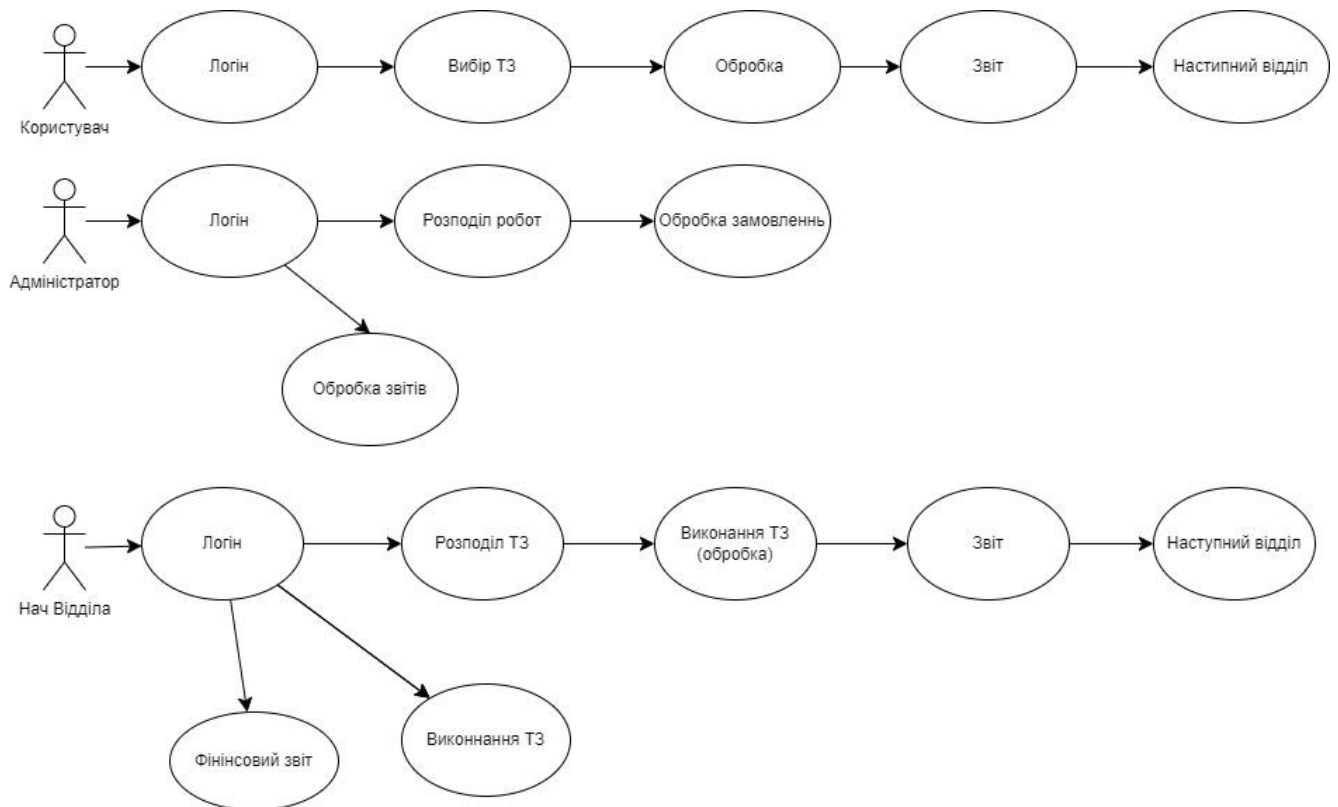


Рисунок 2 – Use-case діаграма застосунку (рисунок виконаний самостійно)

Таким чином, архітектура та проектування програмного забезпечення забезпечують надійну основу для розробки ефективної та масштабованої системи, що відповідає вимогам замовника та забезпечує зручність використання.

Користувач може увійти в додаток за допомогою логіну та пароля. У головному меню вибрати один із варіантів роботи, яка раніше була розподілена на нього, для обробки поточного планового завдання. Робота може містити в собі від одного до кількох дизайнів, для однієї або різних марок. Адміністратором звітти робота була введена в базу, всі дизайни розкладені по папках, відповідно до внутрішнього алгоритму розподілу та присвоєння робіт.

### 3.1.1 Рівень презентації (UI/UX)

Цей рівень відповідає за взаємодію користувачів із системою. Він реалізований за допомогою Windows Forms, що забезпечує зручний та інтуїтивно зрозумілий інтерфейс користувача. Рівень презентації включає в себе:

- головне вікно програми з меню навігації;
- форми для введення та редагування даних;
- форми для відображення звітів та аналітики.

### 3.1.2 Рівень бізнес-логіки

Рівень бізнес-логіки реалізує основні функціональні можливості системи та забезпечує виконання бізнес-правил. Він включає:

- класи для обробки замовлень, клієнтів, дизайнерів та співробітників;
- логіку розрахунків вартості замовлень;
- логіку генерації звітів.

### 3.1.3 Рівень доступу до даних

Цей рівень відповідає за взаємодію з базою даних. Він реалізований за допомогою Entity Framework, що забезпечує абстрагування від специфіки конкретної СУБД та спрощує роботу з даними. Рівень доступу до даних включає:

- класи контексту бази даних;
- репозиторії для доступу до таблиць бази даних;
- запити до бази даних для отримання, додавання, оновлення та видалення даних.

### 3.1.4 Рівень зберігання даних

Рівень зберігання даних відповідає за фізичне зберігання даних у базі даних. Для цього проекту обрана СУБД Microsoft SQL Server, що забезпечує високу продуктивність, надійність та масштабованість. База даних містить таблиці для зберігання інформації про замовлення, клієнтів, співробітників та дизайнерів.

### 3.2 Проектування архітектури

Проектування архітектури включає створення діаграм, що ілюструють структуру системи та взаємодію між її компонентами. Для цього використовуються діаграми UML. В нашій системі ми використали наступне рішення. На рисунку 3 нижче, вказана UML діаграма, завдяки якій ми розробляли нашу систему.

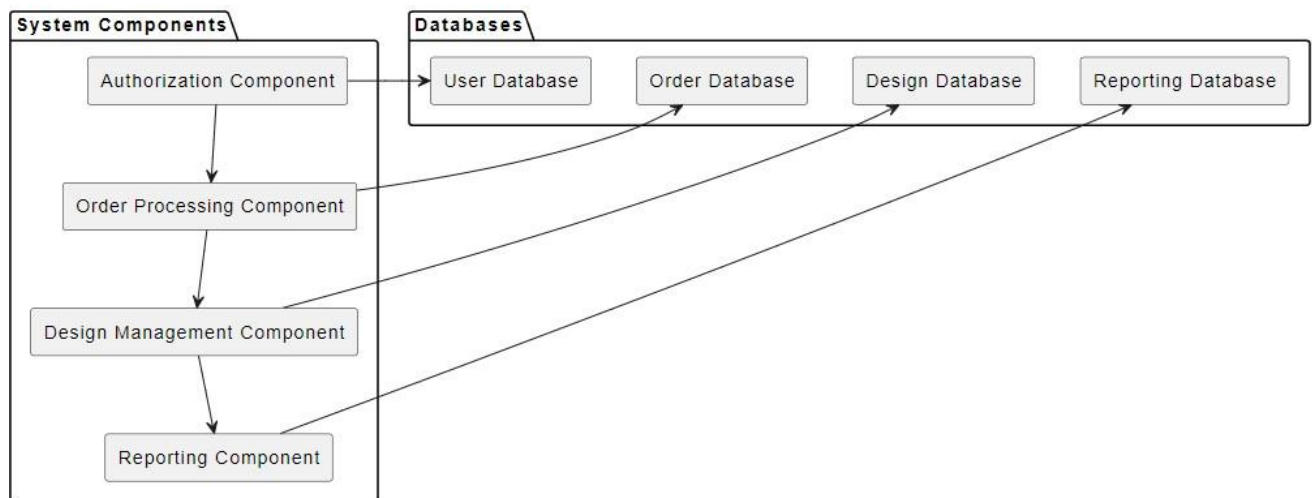


Рисунок 3 – UML діаграма застосунку (рисунок виконаний самостійно)

#### 3.2.1 Діаграма класів

Діаграма класів відображає основні класи системи та їх взаємозв'язки. Основні класи включають:

- клас Order для зберігання інформації про замовлення;
- клас Client для зберігання інформації про клієнтів;

- клас Designer для зберігання інформації про дизайнерів;
- клас Employee для зберігання інформації про співробітників.

### 3.2.2 Діаграма компонентів

Діаграма компонентів відображає основні компоненти системи та їх взаємодію. Основні компоненти включають:

- компонент інтерфейсу користувача;
- компонент бізнес-логіки;
- компонент доступу до даних;
- компонент бази даних.

### 3.2.3 Діаграма розгортання

Діаграма розгортання відображає фізичне розміщення компонентів системи на різних серверах та робочих станціях. Основні елементи включають:

- сервер бази даних;
- сервер застосунків;
- клієнтські робочі станції.

## 3.3 Проектування структури зберігання даних

Проектування структури зберігання даних включає розробку схеми бази даних. Схема бази даних включає:

- таблиці для зберігання інформації про замовлення, клієнтів, дизайнерів та співробітників;
- зв'язки між таблицями;
- індекси для оптимізації запитів;
- тригери та процедури для забезпечення цілісності даних.

## 3.4 Приклади найцікавіших алгоритмів та методів

### 3.4.1 Алгоритм розрахунку вартості замовлення

Цей алгоритм використовується для автоматичного обчислення кінцевої вартості замовлення на основі вибраних продуктів, їх кількості та застосованих знижок. Він також враховує додаткові послуги, такі як швидка доставка чи особливе пакування та представляється або як рисунок, або як лістинг.

```
public decimal CalculateOrderCost(List<OrderItem> items, Customer
customer) {
    decimal totalCost = 0;
    foreach (var item in items) {
        decimal itemCost = item.Quantity * item.Product.Price;
        if (customer.IsEligibleForDiscount()) {
            itemCost -= itemCost * customer.DiscountRate;
        }
        totalCost += itemCost;
    }
    if (customer.SelectedService.Contains("ExpressDelivery")) {
        totalCost += 20; // Припустимо, що експрес-доставка коштує
        додатково 20 одиниць валюти
    }
    return totalCost;
}
```

Цей алгоритм дозволяє системі обробки замовлень ефективно розраховувати вартість з урахуванням індивідуальних параметрів кожного клієнта та забезпечує гнучкість ціноутворення.

### 3.4.2 Алгоритм генерації щомісячних звітів

Цей алгоритм автоматизує процес створення звітів про діяльність дизайн-студії, збираючи дані про всі замовлення, реалізовані протягом місяця, та аналізуючи загальний дохід, популярність продуктів, та інші ключові показники.

```
public MonthlyReport GenerateMonthlyReport(DateTime month) {
    var orders = dbContext.Orders.Where(o => o.DateCreated.Month ==
month.Month && o.DateCreated.Year == month.Year);
    var report = new MonthlyReport() {
        TotalIncome = orders.Sum(o => o.TotalCost),
        TotalOrders = orders.Count(),
        MostPopularProducts = orders
            .SelectMany(o => o.Items)
            .GroupBy(i => i.ProductId)
            .OrderByDescending(g => g.Count())
            .Take(3)

        .Select(g => g.Key).ToList()
    };
    return report;
}
```

}

Цей алгоритм надає важливу інформацію для власників бізнесу, допомагаючи у прийнятті рішень щодо маркетингу, виробничих потужностей та стратегії розвитку.

Ці приклади демонструють, як комплексні алгоритми інтегровані у систему для поліпшення її ефективності та забезпечення важливого функціоналу, необхідного для сучасної дизайн-студії.

### 3.5 Створення UI/UX

Проектування користувацького інтерфейсу включає:

- розробку макетів основних форм програми;
- використання принципів зручності та інтуїтивності інтерфейсу;
- забезпечення доступності та ергономіки інтерфейсу.

Таким чином, у даному розділі ми розглянули основні аспекти архітектури та проектування нашого програмного забезпечення. Завдяки багаторівневій архітектурі можна чітко розділити відповідальності між різними шарами системи, що сприяє кращому управлінню кодом та полегшує його подальше розширення та обслуговування.

Модуль авторизації забезпечує безпечний вхід у систему, дозволяючи користувачам доступ до їхніх замовлень та дизайнів відповідно до присвоєних прав. Модуль керування дизайнами реалізує всі необхідні операції для роботи з дизайнерськими проектами, включаючи їх створення, редагування та видалення. Рівень зберігання даних надійно управляє всією інформацією, яка зберігається у базі даних, забезпечуючи її цілісність та безпеку.

Застосування UML діаграм значно поліпшило процес проектування, оскільки надало можливість наочно представити всі компоненти системи та їх взаємодії.

У результаті впровадження розглянутих архітектурних підходів ми отримали масштабовану, гнучку та надійну систему, що відповідає всім сучасним вимогам

до програмного забезпечення і забезпечує високу якість обслуговування користувачів.

## 4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

### 4.1 Створення програмного додатка

У даному розділі ми описуємо програмні рішення, які були прийняті під час розробки інформаційної системи для обробки замовлень дизайн-студії поліграфічного підприємства. Перед вибором конкретних рішень, ми провели аналіз вимог до системи та врахували особливості бізнес-процесів компанії.

Мова програмування C# була обрана через її потужні об'єктно-орієнтовані можливості та розширену типізацію, що є критично важливими для створення надійних додатків. Вона підтримує багатопотоковість, збирання сміття, виключення типів, що спрощує обробку помилок, та має розгалужену систему бібліотек для швидкого доступу до різноманітних функціональних можливостей. Це дозволяє розробникам зосередитися на бізнес-логіці додатку, мінімізуючи зусилля на низькорівневе програмування.

Windows Forms є ідеальним вибором для створення десктопних застосунків у середовищі .NET. Ця технологія дозволяє розробляти графічні користувацькі інтерфейси шляхом простого перетягування компонентів на форму, забезпечуючи високий рівень абстракції від більш складних аспектів програмування інтерфейсу. Такий підхід не тільки прискорює процес розробки, але й гарантує, що розробники можуть легко модифікувати та адаптувати інтерфейс до мінливих вимог користувачів.

Додатково, Windows Forms підтримує велику кількість стандартних контролів, таких як текстові поля, кнопки, лейбли, меню, які можуть бути легко налаштовані під користувача та розширені через наслідування. Це забезпечує не тільки функціональність, але й гнучкість в дизайні користувацького інтерфейсу. Візуальна консистентність та інтеграція з операційною системою Windows забезпечують, що кінцеві користувачі

отримують знайомий та зручний досвід використання, що є ключовим для комерційних застосунків, де зручність користувача є пріоритетом.

Нижче представлено рисунок 4, що до інтерфейсу нашого програмного забезпечення

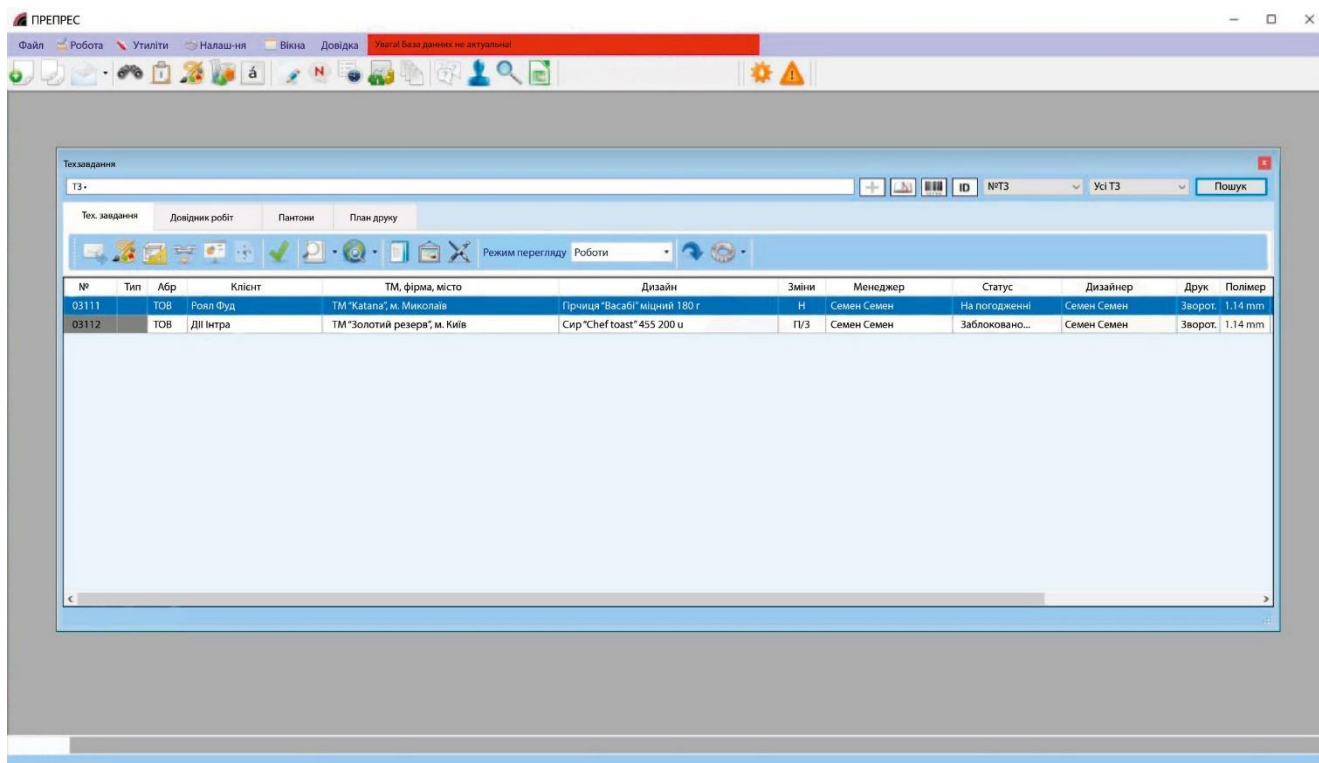


Рисунок 4 – Приклад інтерфейсу.(Виконано самостійно)

Програмне забезпечення має дуже простий вигляд. Мінімально необхідній для використання. Наступний рисунок 5 показує інформацію що до інтерфейсу технічного завдання

Техничне завдання №03331

Основні дані    Дизайни    Збірки (схема)    Технологічні дані

Управління роботою в ТЗ

Робота

Замовник: ТОВ ДІІ Інтра, ТМ "Золотий резерв", м. Київ

Робота: Сир "Chef toast" 45% 200 г

Обрати

Дані з еталону    Дані з бланку    Дані по штрихкоду    Скинути

Параметри

Шаг (мм) 210    Тип зміни П/3    № бланку 0    Відкрити    4820200488301

Ширина (мм) 330    Вид упаковки Пакет    Служ. інф.    3 екрану

Кільк. полос 3    Полімер 1,14 мм    Профіль 2017\_obr\_PPpr+W+MPP\_140lpIDc

4 820200 488301

Коментарі

Зміни

Технолога

Дизайнера

Cyan	C	•	Kodak	Maxtone CX - 30
Magenta	M	•	Kodak	Maxtone CX - 30
Yellow	Y	•	Kodak	Maxtone CX - 30
Black	K	•	Kodak	Maxtone CX - 30
Black 6	B	•	Kodak	Maxtone CX - 30
White	W	•	Kodak	Maxtone CX - 30
872	G	•	Kodak	Maxtone CX - 30
460	P1	•	Kodak	Maxtone CX - 30

Кольор. група (8/0/8)

Код роботи 18.3ED.3C9B.DFC2

Назначити зображення

Палітра кольорів:  без комп.  комп 2017  комп. 2019  Крива 140

Папка роботи    Папка для збірки    Назва файлу    Файл схеми    Папка дизайну    Старий шлях

ДОДАТИ НОВУ РОБОТУ    Редагувати

Множинне виділення    Зберігати для усіх виділень:  Профіль  Кольорову групу  Прапор компенсації  Прапор кривої

ID	Замовник	Зміни	Дизайн	Дата
57282	ТОВ ДІІ Інтра, ТМ "Золотий резерв", м. Київ	П/3	Сир "Chef toast" 45% 200 г	09/10/2




Рисунок 5 – Приклад інтерфейсу.(Виконано самостійно)

Ми можемо побачити назву дизайну, розмір, кольори, що будуть в роботі, принцип виводу файлів, також кнопки для переходу до каталогів, де зберігаються необхідні файли, та інше.

Що до вибору бази даних, де ми планували зберігати інформацію. Для зберігання даних ми обрали систему управління базами даних Microsoft SQL Server. Використання SQL Server дозволило нам забезпечити надійне зберігання та ефективний доступ до даних, а також використовувати різноманітні можливості для оптимізації запитів та забезпечення безпеки даних.

Microsoft SQL Server є потужною системою управління базами даних, яка чудово підходить для корпоративних додатків, де висувуються високі вимоги до продуктивності, надійності, та безпеки даних. Його вибір як основи для нашої системи обробки замовлень був обумовлений кількома ключовими чинниками.

SQL Server забезпечує високу продуктивність завдяки своїм вбудованим інструментам для оптимізації запитів, таким як індексація та паралелізм запитів. Це дозволяє системі ефективно обробляти великі обсяги даних та складні запити, що є критично важливим для забезпечення швидкої відповіді на запити користувачів в умовах високого навантаження.

Він підтримує широкий спектр опцій транзакційної консистентності, які забезпечують надійне управління даними. Використання транзакцій дозволяє гарантувати, що всі операції з даними в системі є атомарними, послідовними, ізольованими та стійкими (ACID), що мінімізує помилки та забезпечує високу якість даних.

SQL Server надає розширені можливості для налаштування безпеки, включаючи шифрування даних, аудит, та управління доступом на рівні рядків. Це дозволяє забезпечити захист конфіденційної інформації та контролювати доступ до даних з максимальною гнучкістю.

Завдяки своїм можливостям горизонтального та вертикального масштабування, SQL Server відмінно підходить для систем, що ростуть разом із зростанням підприємства. Можливість додавання додаткових серверів або ресурсів для обробки збільшеного обсягу даних забезпечує плавне розширення без перерви у роботі системи.

Використовуючи SQL Server, ми створили реляційну базу даних з таблицями для управління замовленнями, клієнтами, товарами та іншими пов'язаними даними. На рисунку 6, нижче представлена діаграма таблиць нашої бази даних.

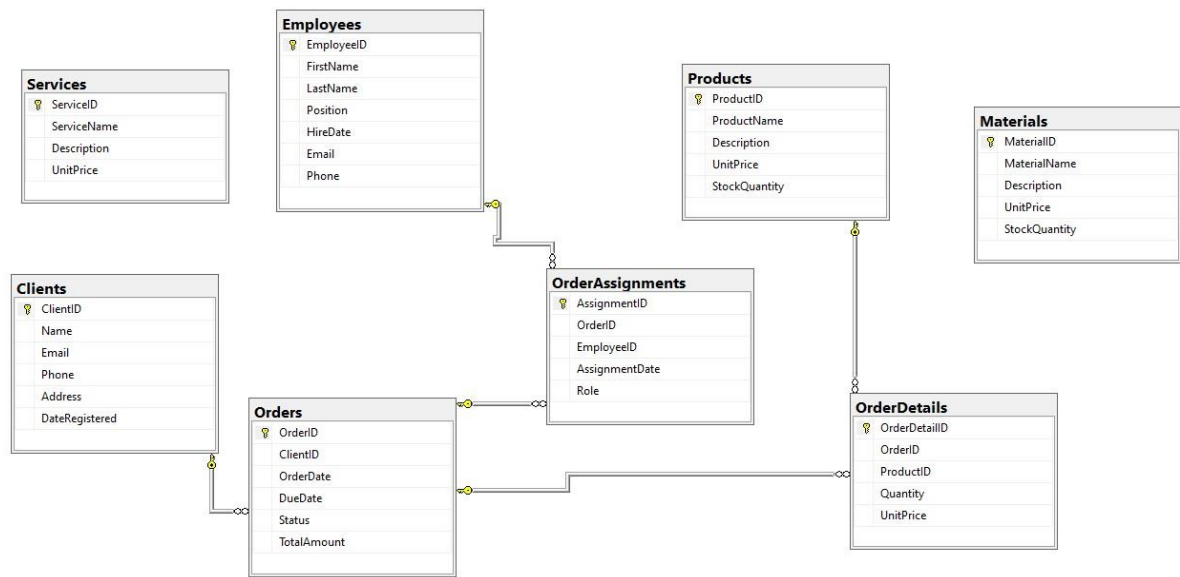


Рисунок 6 – Приклад діаграми бази даних.(Виконано самостійно )

Ця структура бази даних забезпечуватиме ефективне управління клієнтами, замовленнями, матеріалами, послугами та персоналом.

Впровадження таких рішень дозволяє не тільки оптимізувати роботу з даними, але й підвищити ефективність загальної роботи системи, забезпечуючи високу швидкість відгуку та надійність управління замовленнями.

Для організації програмного коду ми використовували принципи об'єктно-орієнтованого програмування (ООП). Код був структурований на класи, що дозволило розділити функціональність додатку на логічні блоки та забезпечити його модульність та розширюваність.

Використовувалися кілька основних принципів об'єктно орієнтованого програмування. по перше інкапсуляція.

Інкапсуляція дозволила обмежити доступ до деяких компонентів класу, що сприяло кращому контролю над даними і зменшило можливість помилок. Наприклад, ми заховали внутрішні структури даних класу Order

і надавали доступ до них тільки через методи, які виконували необхідні перевірки даних.

В якості прикладу, на рисунку 7 нижче представлено фрагмент коду, в якому ми використовуємо ці принципи.

```
1 using System;
2
3 public class Order
4 {
5     private int orderId;
6     private string customerName;
7     private DateTime orderDate;
8     private string status;
9     // Конструктор для ініціалізації замовлення
10    public Order(int orderId, string customerName, DateTime orderDate, string status)
11    {
12        this.orderId = orderId;
13        SetCustomerName(customerName);
14        this.orderDate = orderDate;
15        this.status = status;
16    }
17    // Властивість для доступу та встановлення ID замовлення
18    public int OrderId
19    {
20        get { return orderId; }
21        set { orderId = value; }
22    }
23    // Метод для встановлення імені клієнта з перевіркою на порожнє значення
24    public void SetCustomerName(string name)
25    {
26        if (!string.IsNullOrEmpty(name))
27        {
28            customerName = name;
29        }
30        else
```

Рисунок 7 – Приклад коду конструктора замовлень. (Виконано самостійно)

Цей клас Order демонструє використання інкапсуляції в C#, де внутрішні дані (наприклад, customerName та status) захищені від прямого доступу ззовні. Замість цього використовуються методи SetCustomerName та UpdateStatus, які включають перевірки для забезпечення валідності даних перед їх встановленням. Це допомагає забезпечити більшу контроль над тим, як дані використовуються та модифікуються, знижуючи ризик помилок чи некоректного використання об'єктів класу.

Друге, що ми використовували це поліморфізм.

Поліморфізм дозволив об'єктам виконувати операції через однакові інтерфейси з різною поведінкою. Це особливо корисно при розширенні функціональності програми, дозволяючи використовувати нові класи без зміни існуючого коду, що взаємодіє з базовими класами.

Також, спадкування. Спадкування дозволило розробникам створювати нові класи на основі існуючих, розширюючи існуючі функціональності і водночас зберігаючи код чистим та легким для розуміння. Це також сприяє уникненню дублікації коду.

ООП дозволяє програмістам моделювати реальні сценарії, використовуючи об'єкти та класи, що відповідають елементам бізнес-процесів, що робить код більш інтуїтивно зрозумілим і зменшує помилки у взаємодії з програмою.

Кожен з цих аспектів ООП відіграє важливу роль у створенні масштабованої, підтримуваної та ефективної програмної системи. Такий підхід не тільки спрощує розробку та подальше внесення змін до програми, але й забезпечує високий рівень адаптивності та відповідності до змінних вимог користувачів.

Організація коду з використанням принципів об'єктно-орієнтованого програмування (ООП) має критичне значення для створення чистого, модульного та легко підтримуваного програмного забезпечення. В ООП класи та об'єкти є основними будівельними блоками, де класи визначають шаблони атрибутів (даних) та поведінок (методів), а об'єкти є екземплярами класів.

У рамках даної роботи ми зосередилися на розробці програмного забезпечення для автоматизації обробки замовлень у дизайн-студії поліграфічного підприємства. Актуальність проекту визначається необхідністю підвищення ефективності роботи студії, оптимізації процесів обробки інформації про замовлення та вдосконалення взаємодії з клієнтами.

Обрані технології мають активну спільноту та регулярні оновлення, що забезпечує тривалу підтримку та безпеку програмного продукту. Вибрані технології відповідали рівню знань та досвіду розробників, що забезпечило високу швидкість розробки та якість кінцевого продукту. Технології які ми використовували, дозволяють масштабувати проект у відповідності з зростаючими потребами підприємства.

Архітектура системи була розроблена з використанням багаторівневого підходу, що включала розділення відповідальностей між компонентами.

Перше, це презентаційний рівень, що відповідає за взаємодію з користувачем та відображення необхідної інформації. Бізнес-логіка забезпечила обробку даних та реалізацію ключових процесів системи.

Доступ до даних ізолює бізнес-логіку від деталей реалізації зберігання даних, що спрощує модифікацію та масштабування системи. Ці рішення лягли в основу розробки надійної та ефективної системи, здатної задовольнити потреби сучасного поліграфічного підприємства. Далі ми спробуємо детально описати кожен аспект прийнятих програмних рішень, від архітектури до конкретних.

Давайте більш детально розглянемо архітектурні компоненти системи.

Авторизація користувачів є критичною частиною забезпечення безпеки системи. Модуль авторизації реалізовано з використанням стандартних практик безпеки, таких як шифрування паролів та сесійний менеджмент. Основна мета модуля полягає в перевірці користувацьких даних на відповідність збереженим у базі даних, а також управлінні сесіями користувачів для запобігання несанкціонованого доступу.

Сесії користувачів керуються через токени доступу, які генеруються після успішної авторизації та зберігаються у веб-клієнті. Сервер перевіряє цей токен з кожним запитом на автентичність та актуальність.

Модуль обробки замовлень є основним компонентом системи, оскільки він відповідає за управління всім циклом замовлення — від його створення до виконання. Цей модуль інтегрований з модулями управління клієнтами та дизайнами, що забезпечує цілісний підхід до обробки замовлень.

Модуль управління дизайнами. Цей модуль відповідає за зберігання, редагування та видалення дизайнів, які використовуються у замовленнях. Він дозволяє дизайнерам ефективно керувати своїми проектами, відстежувати версії та впроваджувати зміни без ризику втрати даних. Функціональність версіонування. Кожне збереження дизайну створює нову версію в системі, дозволяючи користувачам переглядати історію змін та, при необхідності, відновлювати попередні версії.

У процесі розробки програмного забезпечення було прийнято стратегічне рішення використовувати набір технологій, що добре відомі своєю надійністю, гнучкістю та широкою підтримкою. Ключові технології, що були обрані для проекту, описані нижче.

Microsoft .NET Framework: Платформа була вибрана через свою стабільність, велику підтримку спільноти та багатий набір інструментів для розробки. .NET відомий своєю високою продуктивністю та інтеграцією з багатьма базами даних та веб-сервісами.

C# як основна мова програмування: C# обрано за його потужні об'єктно-орієнтовані можливості, чудову підтримку бібліотек та інтеграцію з .NET Framework.

Вибір Windows Forms був зумовлений потребою у швидкій розробці традиційних десктопних додатків з багатим користувацьким інтерфейсом.

Microsoft SQL Server як система управління базами даних (СУБД): SQL Server використовувався через його високу продуктивність, надійність, підтримку транзакцій та легкість інтеграції з .NET додатками.

Цей опис підкреслює як технічну глибину прийнятих рішень, так і їх практичне застосування, виходячи з потреб проекту, що є ключовим для повноцінного розуміння реалізації і значення кваліфікаційної роботи.

Розроблене програмне забезпечення це система, яка забезпечує ефективно управління замовленнями. Автоматизація процесів від приймання замовлення до його виконання, знижує помилки, пов'язані з людським фактором.

Інструменти для керування дизайн-проектами, дозволять краще координувати роботу дизайнерів і стежити за виконанням замовлень. Швидке реагування на запити клієнтів та можливість надання персоналізованих послуг завдяки глибшому аналізу даних клієнтів.

Архітектура нашого програмного забезпечення дозволяє легко масштабувати систему для обробки збільшеного обсягу замовлень без потреби у значних змінах коду. Використання сучасних практик безпеки захищає персональні та фінансові дані клієнтів. Система побудована з можливістю легкої інтеграції з іншими інструментами і технологіями, що може знадобитися в майбутньому.

Прийняті програмні рішення повністю відповідають поставленим завданням розробки. Використання .NET та SQL Server забезпечило високу продуктивність та надійність системи, а вибір C# та Windows Forms дозволив реалізувати багатий і зрозумілий користувацький інтерфейс. Рішення були прийняті з урахуванням поточних та майбутніх потреб бізнесу, що забезпечує тривалу вартість інвестицій в розробку.

## 5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 5.1 Тестування застосунку

У даному розділі ми описуємо підходи до тестування розробленого програмного забезпечення для інформаційної системи обробки замовлень дизайн-студії поліграфічного підприємства. Тестування є важливою складовою частиною розробки програмного забезпечення, оскільки дозволяє перевірити його працездатність, надійність та відповідність вимогам замовника.

Одним з видів тестування програмного забезпечення є функціональне тестування.

Проведення функціонального тестування дозволяє перевірити, чи виконуються всі функціональні вимоги до системи. Ми розробили набір тест-кейсів, які охоплюють основні функціональні можливості системи, такі як прийом та обробка замовлень, робота з базою даних, генерація звітів тощо. Кожен тест-кейс був виконаний для перевірки правильності роботи відповідної функціональності наведено в таблиці 5.1.

Таблиця 5.1 - Функціональне тестування Windows Forms застосунку

Тест-кейс ID	Опис функції	Кроки виконання	Очікуваний результат	Фактичний результат	Примітка
ТС-01	Прийом нового замовлення	1. Відкрити форму нового замовлення. 2. Ввести деталі замовлення.	Замовлення зберігається у базі даних і відображається у	З'являється нова запис в базі даних ТЗ	

Тест - кейс ID	Опис функції	Кроки виконання	Очікуваний результат	Фактичний результат	Примітка
		3. Натиснути кнопку "Зберегти".	відповідному інтерфейсі.		
ТС-02	Перевірка обов'язкових полів	1. Відкрити форму нового замовлення. 2. Залишити обов'язкові поля порожніми. 3. Натиснути кнопку "Зберегти".	Система відображає помилку про незаповнені обов'язкові поля.	З'являється вікно з текстом «Помилка вводу»	

Для перевірки працездатності системи під навантаженням було проведено навантажувальне тестування. Ми моделювали ситуації, коли система одночасно отримує велику кількість запитів на обробку замовлень та інших операцій. Під час цього тестування ми вимірювали час відповіді системи та переконувалися, що вона в змозі ефективно працювати при великому обсязі даних, наведено в таблиці 5.2.

Таблиця 5.2 - Навантажувальне тестування системи

Тест-кейс ID	Опис функції	Кроки виконання	Очікуваний результат	Фактичний результат	Примітка
ТС-03	Тестування системи під високим навантаженням	1. Симулювати одночасне введення даних 100 користувачами. >2. Моніторити час відповіді системи.	Система ефективно обробляє запити без помилок та затримок.	Програма реагує на діяльність користувача	

Інший вид тестування – тестування безпеки.

Оскільки наша система містить конфіденційну інформацію про клієнтів та їх замовлення, ми приділили особливу увагу тестуванню безпеки. Ми перевірили систему на наявність потенційних вразливостей, таких як SQL-ін'єкції, перехоплення сесій тощо, і внесли необхідні заходи для їх усунення наведено в таблиці 5.3.

Таблиця 5.3 - Тестування безпеки

Тест-кейс ID	Опис функції	Кроки виконання	Очікуваний результат	Фактичний результат	Примітка
ТС-04	Перевірка системи на SQL-ін'єкції	1. Ввести SQL команди в поля введення даних. >2. Натиснути "Зберегти" або	Система не виконує невалідні SQL команди, дані залишаються безпечними.	Вивід вікна «Дані оновлені»	

Тест-кейс ID	Опис функції	Кроки виконання	Очікуваний результат	Фактичний результат	Примітка
		іншу кнопку, що викликає обробку даних.			

Наступний від тестування сумісності.

Оскільки наша програмна система призначена для роботи на різних пристроях та під різними операційними системами, ми провели тестування сумісності. Ми перевірили, як система працює на різних версіях Windows та в різних браузерах, забезпечуючи її стабільну роботу на різних платформах наведено в таблиці 5.4.

Таблиця 5.4 - Тестування сумісності

Тест-кейс ID	Опис функції	Кроки виконання	Очікуваний результат	Фактичний результат	Примітка
ТС-05	Сумісність з різними версіями Windows	1. Відкрити систему на Windows 7, 8, 10 та 11. 2. Перевірити основні функції (наприклад, створення замовлення).	Система коректно функціонує на всіх перевірених версіях Windows.	Програмне забезпечення запускається під οποю ОС	

За результатами проведеного тестування були виявлені та усунуті різноманітні помилки та недоліки, що дозволило забезпечити високу якість та надійність розробленого програмного забезпечення.

## 6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 6.1 Визначення плану впровадження

Впровадження програмного забезпечення – це критичний етап, який визначає успішність його подальшої експлуатації. Для правильного формування етапу застосування онлайн-системи необхідно скласти детальний план, що допоможе уникнути помилок та забезпечити плавний перехід до нової системи.

План впровадження складається з кількох основних етапів, які охоплюють підготовчі дії, встановлення, тестування, навчання персоналу та введення системи в експлуатацію.

#### Етапи впровадження програмного забезпечення

Для більшої наочності етапи впровадження представлені у вигляді таблиці 6.1.

Таблиця 6.1 – Етап впровадження

Етап	Опис	Відповідальні особи	Термін виконання
1. Підготовчий етап	Аналіз поточних процесів, визначення вимог, складання плану впровадження	Керівник проекту, аналітики	2 тижні
2. Встановлення ПЗ	Інсталяція програмного забезпечення на сервери та робочі станції	Системні адміністратори	1 тиждень
3. Міграція даних	Перенесення існуючих даних до нової системи, перевірка коректності міграції	Системні адміністратори, розробники	1 тиждень

Етап	Опис	Відповідальні особи	Термін виконання
4. Тестування системи	Функціональне та навантажувальне тестування, виявлення та виправлення помилок	Тестувальники, розробники	2 тижні
5. Навчання персоналу	Проведення тренінгів для користувачів системи, створення документації та інструкцій	Тренери, аналітики	1 тиждень
6. Пілотне впровадження	Впровадження системи в окремому підрозділі для тестування в реальних умовах	Керівник проекту, користувачі	2 тижні
7. Виправлення недоліків	Збір відгуків, аналіз проблем, внесення змін та вдосконалень до системи	Розробники, тестувальники	1 тиждень
8. Повне впровадження	Запуск системи в експлуатацію в усіх підрозділах підприємства	Керівник проекту, користувачі	1 тиждень
9. Підтримка та оновлення	Постійна технічна підтримка користувачів, регулярні оновлення та вдосконалення системи	Розробники, підтримка	Постійно

Краткий опис по кожному пункту що до таблиці.

Підготовчий етап. На цьому етапі проводиться детальний аналіз існуючих бізнес-процесів підприємства та визначення вимог до нової системи. Складається детальний план впровадження з визначенням відповідальних осіб та термінів виконання кожного етапу.

Встановлення ПЗ. Встановлення програмного забезпечення на сервери та робочі станції підприємства. Встановлюються всі необхідні компоненти, проводяться налаштування та перевірка коректності роботи системи.

Міграція даних. Перенесення існуючих даних до нової системи. Включає експорт даних зі старих систем, трансформацію та завантаження в нову базу даних. Проводиться перевірка коректності міграції.

Тестування системи. Проводиться функціональне тестування для перевірки відповідності системи вимогам. Виконується навантажувальне тестування для оцінки продуктивності та виявлення потенційних вузьких місць. виправляються виявлені помилки.

Навчання персоналу. Організуються тренінги для користувачів системи. Створюються детальні інструкції та документація, що допоможуть персоналу швидко адаптуватися до роботи з новою системою.

Пілотне впровадження. Впровадження системи в окремому підрозділі підприємства для тестування в реальних умовах. Збір відгуків користувачів та аналіз результатів.

Виправлення недоліків. На основі відгуків від пілотного впровадження вносяться зміни та вдосконалення до системи. Проводиться додаткове тестування для перевірки виправлених помилок.

Повне впровадження. Запуск системи в експлуатацію в усіх підрозділах підприємства. Проводиться моніторинг роботи системи та надання підтримки користувачам.

Підтримка та оновлення. Після впровадження системи здійснюється постійна технічна підтримка користувачів. Регулярно випускаються

оновлення та вдосконалення для забезпечення безперебійної роботи системи та її відповідності новим вимогам.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи була розроблена програмна система обробки замовлень для дизайн-студії поліграфічного підприємства. В результаті дослідження та розробки були досягнуті значні результати. Спочатку був проведений детальний аналіз бізнес-процесів поліграфічного підприємства, що дозволило виявити ключові проблеми та потреби, які можна вирішити за допомогою автоматизації. Було визначено основні функції та вимоги до системи, які відповідають потребам дизайн-студії.

На основі аналізу предметної області та потреб користувачів були сформульовані технічні вимоги до системи. Ці вимоги охоплювали функціональні аспекти, такі як управління замовленнями, збереження інформації про клієнтів та замовлення, генерація звітів, а також нефункціональні аспекти, включаючи безпеку, продуктивність та зручність користування.

Архітектура програмної системи була розроблена на основі Windows Forms та реляційної бази даних SQL Server. Використання сучасних підходів та технологій, таких як об'єктно-орієнтоване програмування, патерни проектування та інтеграція з SQL Server для збереження даних, дозволило створити надійну систему. Буде створено зручний та інтуїтивно зрозумілий інтерфейс користувача, який дозволяє легко взаємодіяти з системою.

Після розробки системи буде проведено ретельне тестування, включаючи функціональні, інтеграційні та регресійні тести. Виявлені помилки були виправлені, що забезпечить високу якість та надійність системи. Розроблений план впровадження системи включав підготовку інструкцій для користувачів та адміністраторів, а також проведення навчальних сесій.

Впровадження системи дозволить значно покращити ефективність та точність обробки замовлень, зменшити кількість помилок та підвищити задоволеність клієнтів. Таким чином, розроблена програмна система буде повністю відповідати поставленим цілям та вимогам. Вона забезпечить автоматизацію ключових бізнес-процесів дизайн-студії поліграфічного підприємства, покращує продуктивність та якість роботи, а також підвищить задоволеність клієнтів.

Подальший розвиток системи може включати інтеграцію з іншими корпоративними системами, розширення функціональності та адаптацію до нових потреб підприємства.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Електронний ресурс. – Режим доступу: <https://www.microsoft.com/en-us/sql-server/sql-server-2022> – Назва з екрану. – Дата звернення: 19.06.2024.
2. 1С: Підприємство. [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki/1%D0%A1:%D0%9F%D1%96%D0%B4%D0%BF%D1%80%D0%B8%D1%94%D0%BC%D1%81%D1%82%D0%B2%D0%BE>. Дата звернення: 08.06.2024.
3. SAP. (2024). Рішення для управління взаємовідносинами з клієнтами (CRM) від SAP [Електронний ресурс]. Отримано з <https://www.sap.com/ukraine/products/crm/solutions.html>
4. Bitrix 24. [Електронний ресурс]. URL: <https://www.bitrix24.com/>. Дата звернення: 08.06.2024.
5. Офіційний сайт MySQL. [Електронний ресурс]. URL: <https://dev.mysql.com/doc/>. Дата звернення: 08.06.2024.
6. Clean Architecture. [Електронний ресурс]. URL: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>. Дата звернення: 08.06.2024.
7. Database Design - Tutorial by TutorialsPoint. [Електронний ресурс]. URL: [https://www.tutorialspoint.com/dbms/database\\_design.htm](https://www.tutorialspoint.com/dbms/database_design.htm). Дата звернення: 08.06.2024.
8. Troelsen A., Japikse P. Pro C# 8.0 and the .NET Core 3.0 Framework. Apress, 2019. 1008 p.
9. Redgate. SQL Server Execution Plans. Redgate Books, 2018. 288 p.
10. Petzold C. Programming Windows Forms with C#. Microsoft Press, 2005. 1248 p.
11. McLaughlin M., Van der Lans R., Mettler M. MySQL & MariaDB: A Practical Guide. Addison-Wesley Professional, 2015. 288 p.

12. Albahari J., Albahari B. C# 8.0 in a Nutshell: The Definitive Reference. O'Reilly Media, 2020. 1056 p.
13. Freeman A. Pro ASP.NET Core 3. Apress, 2020. 1080 p.
14. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 560 p.

## ДОДАТОК А

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ




Дата звіту 7/19/2024  
Дата редагування 7/19/2024

Документ прийнятий

---

**метадані**

---

**Заголовок**  
2024\_Б\_ПІ\_ПЗП22-1\_Соколов\_А\_Ю\_скорочений

**Автор** Науковий керівник / Експерт  
Соколов Андрій Юрійович Вадим Юрійович Нечволод


**підприємство**  
Харківський національний університет радіоелектроніки

---

**Тривога**

---

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати намісний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		11
Білі знаки		1
Парафрази (SmartMarks)		16

---

**Обсяг знайдених подібностей**

---

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагиат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

Довжина фраз для коефіцієнта подібності 2

7754

Кількість слів



КЦ

66514

Кількість символів

---

**Подібності за списком джерел**

---

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагиату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз		Копію тексту	
ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	<a href="https://software.nure.ua/wp-content/uploads/2023/05/bachelor_05_05_2023.pdf">https://software.nure.ua/wp-content/uploads/2023/05/bachelor_05_05_2023.pdf</a>	63	0.81 %
2	<a href="https://software.nure.ua/wp-content/uploads/2024/01/dyplom-mag-7_dotm.docx">https://software.nure.ua/wp-content/uploads/2024/01/dyplom-mag-7_dotm.docx</a>	30	0.39 %
3	<a href="https://openarchive.nure.ua/bitstream/document/18694/1/2021_M_PI_Grlnko_KO.pdf">https://openarchive.nure.ua/bitstream/document/18694/1/2021_M_PI_Grlnko_KO.pdf</a>	27	0.35 %
4	<a href="https://software.nure.ua/wp-content/uploads/2023/05/bachelor_05_05_2023.pdf">https://software.nure.ua/wp-content/uploads/2023/05/bachelor_05_05_2023.pdf</a>	25	0.32 %
5	2022_Б_ПІ_ПЗП_18_3_Писаренко_Е_І 5/30/2024 Kharkiv National University of Radio Electronics (Kharkiv National University of Radio Electronics)	24	0.31 %

## ДОДАТОК Б

### Слайди презентації

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра ПІ

1

Тема кваліфікаційної роботи:  
**ІНФОРМАЦІЙНА СИСТЕМА ОБРОБКИ ЗАМОВЛЕНЬ ДИЗАЙН-СТУДІЇ  
ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА.**

Виконав студент  
гр. ПЗПіп-22-1  
Соколов А.Ю.  
Керівник:  
Доцент. Каук В. І.

Харків 2024

## Мета кваліфікаційної роботи

Інформаційна система обробки замовлень дизайн-студії  
поліграфічного підприємства.

2

### У кваліфікаційній роботі розглянуті наступні етапи

- - проведено аналіз вимог для створення повнофункціональної інформаційної системи
- - розробка архітектури програмного забезпечення
- - розробка та тестування програмного забезпечення;

## Мета роботи

3

### ➤ Автоматизація діяльності документообігу поліграфічної студії

- облік клієнтів
- розподіл замовлень між співробітниками
- облік дизайнів
- аналіз даних та статистика
- інтеграція з зовнішніми системами

ХНУРЕ Кафедра ПІ ст. гр. ПЗПіп-22-1 Соколов А. Ю.

## Постановка задачі та опис системи

4



- керівник, який здійснює контроль та управління співробітниками, а також займається формуванням статистичної звітності

- адміністратор, який відповідає за взаємодію з клієнтами, реєстрацію та ведення заявок, роботу з постачальниками

- співробітник, який відповідає за обробку замовлень

ХНУРЕ Кафедра ПІ ст. гр. ПЗПіп-22-1 Соколов А. Ю.

## Обрані засоби реалізації

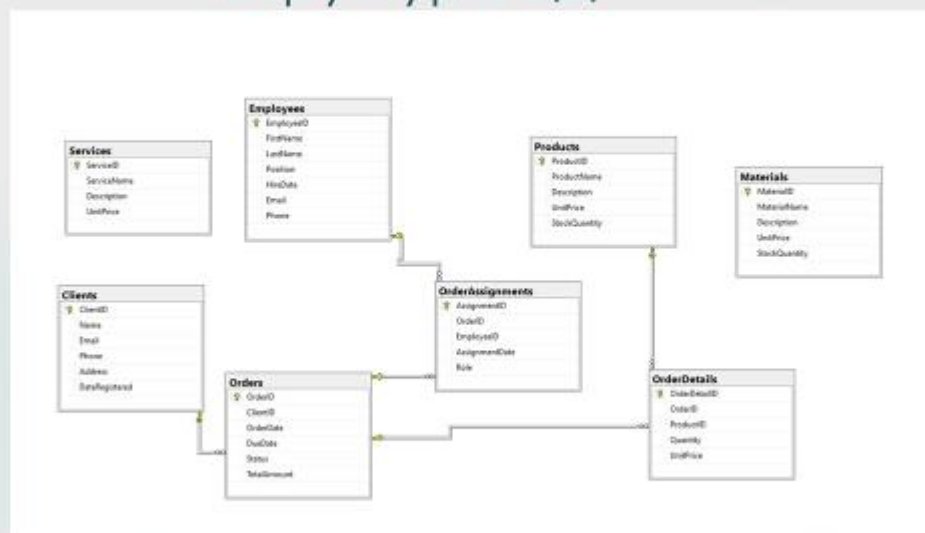
5

- ❑ Microsoft Visual Studio 2022,
- ❑ платформа ASP.Net,
- ❑ мова програмування C#,
- ❑ інтерфейс програмування додатків Windows Forms
- ❑ система управління базами даних Microsoft SQL Server 2022

ХНУРЕ Кафедра ПІ ст. гр. ПЗПп-22-1 Соколов А. Ю.

## Структура БД

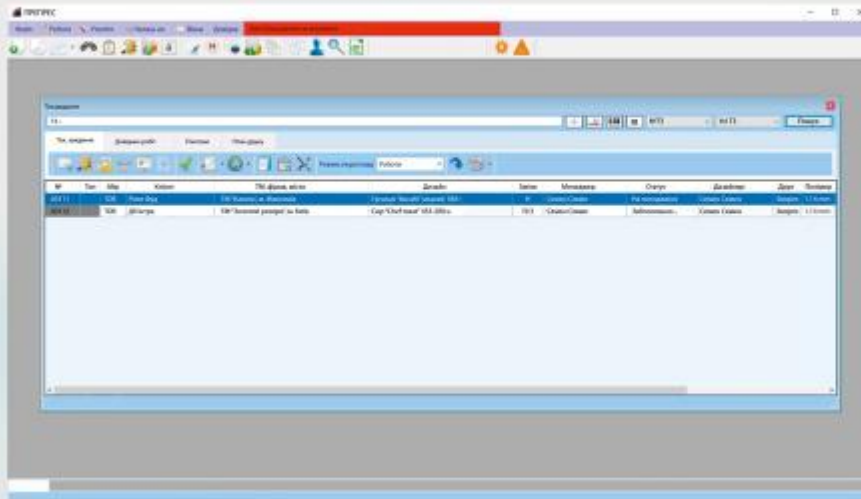
6





## Інтерфейс користувача

9



ХНУРЕ Кафедра ПІ ст. гр. ПЗПіп-22-1 Соколов А. Ю.

## Тестування інформаційної системи

10

- ▶ **Інтеграційне тестування для тестування методів із запитам до БД:**
  - ▶ перевірка кожного методу доступу до БД за участю правильних та неправильних даних;
  - ▶ перевірка БД, щоб переконатися, що дані були заповнені належним чином, чи всі події БД відбулися належним чином;
  - ▶ перевірка сумісності даних із старим застарілим обладнанням та програмним забезпеченням.
- ▶ Критеріями завершення вважалося, коли новий користувач коректно реєструється, є можливість авторизуватися на основі вже існуючих даних з переходом до необхідної пілки доступу, проекти та інші дані коректно додаються, а також є можливість їх редагувати, переглядати та видаляти з БД.
- ▶ **Функціональне тестування:**
  - ▶ виконання кожного сценарію використання або функції з використанням реальних та неправильних даних (валідних та невалідних) для перевірки наступного:
  - ▶ очікувані результати відображаються при використанні дійсних даних;
  - ▶ відповідні повідомлення про помилки або попередження виводяться, коли використовуються невірні дані.
- ▶ **Тестування інтерфейсу користувача.**
  - ▶ Було проведено тестування графічного інтерфейсу користувача – процес тестування продукту інтерфейсу користувача, для забезпечення його відповідності до даної специфікації. Виконувалося для виявлення помилок у функціональності через інтерфейс, необроблені виключення при взаємодії з інтерфейсом, помилки в інтерфейсі.

## Висновки

11

- Проведено аналіз предметної галузі
- Спроектвана архітектура інформаційної системи
- Реалізована та протестована інформаційна система

## ДОДАТОК В

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015