

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматичної, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи і технології»
спеціальності 126 «Інформаційні системи та технології»

на тему: **«Інформаційна система підтримки роботи
будівельного магазину»**

Виконав:

здобувач вищої освіти 4 курсу, групи ІСТ-41
Сех Олександр Богданович

Керівник:

канд. техн. наук, доцент Гладка О. М.

Рецензент:

канд. техн. наук, доцент Барановський С. В.

Національний університет водного господарства та природокористування
ННІ автоматики, кібернетики та обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень – **бакалавр**

за освітньо-професійною програмою **«Інформаційні системи і технології»**

спеціальність **126 «Інформаційні системи та технології»**

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу _____ Сеху Олександрю Богдановичу _____

(прізвище, ім'я, по-батькові)

1. Тема роботи _____ Інформаційна система підтримки роботи будівельного
магазину _____

керівник роботи: _____ Гладка Олена Миколаївна, канд. техн. наук,
доцент _____

(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від “ 19 ” квітня 2023 р. С № 440

2. Термін здачі здобувачем закінченої роботи “ 09 ” червня 2023 р.

3. Вихідні дані до роботи _____ розробити інформаційну систему для обліку роботи
будівельного магазину, що передбачає проектування та створення бази
даних, розробку відповідних форм і функціональних модулів інформаційної
системи засобами Windows Forms і
C#. _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить
розробити) _____ вступ; аналіз предметної області; вибір та обґрунтування
засобів розробки програмного продукту; розробка і опис створеної

інформаційної

системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1.	Гладка О. М., доцент		
2.	Гладка О. М., доцент		
3.	Гладка О. М., доцент		

7. Дата видачі завдання “_07_” листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз об'єкта дослідження, виявлення існуючих проблем	07.11.22 – 25.11.22	
2.	Аналіз існуючих інформаційних методів (технологій)	28.11.22 – 23.12.22	
3.	Вибір та обґрунтування засобів розробки інформаційної системи	30.01.23 – 24.02.23	
4.	Проектування, розробка та реалізація інформаційної системи	27.02.23 – 30.04.23	
5.	Підготовка тексту кваліфікаційної роботи	10.04.23 – 26.05.23	
6.	Підготовка презентації роботи	29.05.23 – 09.06.23	
7.	Відгук керівника, рецензування роботи, перевірка на плагіат	09.06.23 – 16.06.23	

Здобувач _____

(підпис)

О. Б. Сех

(прізвище і ініціали)

Керівник кваліфікаційної роботи

О. М. Гладка

(підпис)

(прізвище і ініціали)

АНОТАЦІЯ

Сех О.Б. Розробка інформаційної системи підтримки роботи будівельного магазину. Кваліфікаційна робота на здобуття ступеня "бакалавр": 54 с., 42 рис., 2 табл., 1 додаток на 14 стор., 17 літературних джерел.

Об'єктом дослідження є процес управління роботою будівельного магазину.

Предметом дослідження – розробка та запровадження інформаційної системи підтримки будівельного магазину.

У процесі розробки були виконані такі кроки: аналіз предметної області та систем-аналогів, проектування бази даних, розробка користувацького інтерфейсу, реалізація функціоналу системи.

У процесі розробки було використано платформу Windows Forms та мову програмування C#. База даних була реалізована з використанням Access.

Результатом роботи є функціональна інформаційна система, яка дозволяє оптимізувати процеси роботи будівельного магазину, забезпечує швидкий доступ до необхідної інформації, спрощує процес керування продажами, а також допомагає у веденні звітності.

Ключові слова: інформаційна система, будівельний магазин, автоматизація, Windows Forms, C#, база даних Access.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. Аналіз роботи будівельного магазину.	9
1.1. Описання та аналіз об'єкта дослідження.....	9
1.2. Аналіз інформаційних систем аналогів.....	12
РОЗДІЛ 2. Опис моделі та методів її реалізації.	19
2.1. Вимоги до інформаційної системи	19
2.2. Опис моделі інформаційної системи.....	20
2.3. Проектування системи за допомогою UML-діаграм	23
2.4. Вибір технологій розробки.....	31
РОЗДІЛ 3. Опис програмної реалізації інформаційної системи.	37
3.1. Створення бази даних.....	37
3.2. Створення складних запитів у Windows Forms.....	41
3.3. Опис інтерфейсу системи	45
3.4. Опис застосунка.....	52
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТОК А	56

ВСТУП

Інформаційна система стала необхідною складовою в різних сферах діяльності, виконуючи важливу роль. Вона набула великого значення у керуванні бізнесом, державними структурами, освітою, виробництві, управлінні, банківською системою та іншими галузями. Її використовують для збору, обробки, управлінні ресурсами організації, що забезпечує прийняття кращих рішень та високу продуктивність.

Ефективне управління будівельними магазинами стає дуже важливим завданням. Щоб досягти успіху і забезпечити стабільне функціонування підприємств, потрібні ефективні інструменти для підтримки різних процесів, таких як управління товарами, продажами, клієнтами та звітністю.

Однією з проблем є великий обсяг інформації, яку потрібно опрацьовувати, що ускладнює ведення обліку товарів та продажів. Старі методи роботи з документами та ручного обліку вже не відповідають сучасним вимогам ефективності та швидкості роботи. Це призводить до затримок у виконанні продажів, непродуктивного використання ресурсів та погіршення якості обслуговування клієнтів.

Тому розробка та впровадження інформаційної системи підтримки роботи будівельного магазину є дуже важливими та актуальним кроком. Така система допоможе підвищити ефективність управління діяльності підприємства, забезпечити швидкий доступ до необхідної інформації та полегшити процес керування продажами, товарами, звітністю та дозволить покращити якість обслуговування клієнтів.

Метою даної роботи є розробка інформаційної системи підтримки роботи будівельного магазину. Для досягнення цієї мети були поставлені наступні завдання:

1. Аналіз предметної області та систем-аналогів.

2. Проектування бази даних для зберігання інформації про товари, продажі, клієнтів та звітність будівельного магазину.

3. Розробка користувацького інтерфейсу, який буде зручним у використанні системи.

4. Реалізація функціоналу системи з використанням платформи Windows Forms та мови програмування C#.

Об'єктом дослідження є процес управління роботою будівельного магазину. **Предметом дослідження** – розробка та запровадження інформаційної системи підтримки будівельного магазину.

Кваліфікаційна робота складається з трьох розділів, які описують різні аспекти об'єкта дослідження. Перший розділ присвячений аналізу роботи будівельного магазину. Другий розділ розглядає вибір технологій та опис інформаційної системи підтримки будівельного магазину. Третій розділ присвячений опису програмної реалізації інформаційної системи. Також в роботі були використані наукові статті та документація.

РОЗДІЛ 1. Аналіз роботи будівельного магазину.

1.1. Описання та аналіз об'єкта дослідження

Для будівельних магазинів важлива взаємодія з клієнтами та надання якісного обслуговування. Консультацію, допомогу в підборі матеріалів, швидку продаж, доставлення товарів, дає велику ймовірність того що клієнт знову прийде. Насправді ті компанії, які забезпечують високий рівень обслуговування клієнтів, здатні залучити більше клієнтів.

Однією з ключових функцій будівельних магазинів є забезпечення належного контролю і наявності товарів. Вони повинні мати товарів на тижні вперед, щоб задовольняти потреби клієнтів, уникати недостачі товарів. Для того, аби зменшити втрати, оптимізувати витрати та забезпечити швидке постачання товарів в магазин потрібне ефективне керування товарами.

Для успішної роботи будівельного магазину необхідно мати ефективну систему керування будівельного магазину. Автоматизована система управління, електронний облік товарів, замовлень та інші інструменти допомагають покращити ефективність, точність та швидкість обробки процесів у будівельних магазинах. Однак при керуванні будівельним магазином та використанні інформаційних систем часто виникають деякі проблеми.

Одна з важливих проблем з якими стикаються будівельні магазини є керування товарами. Деякі основні аспекти та проблеми керуваннями товарами в будівельних магазинах включають наступне:

- Нестача або велика кількість товарів. Нестача або надмірність товарів можуть мати негативний вплив на діяльність будівельного магазину. Нестача товарів може призвести до втрати клієнтів, продажів. Також багато кількість товарів спричиняють затрати на зберігання, що у підсумку приведе до фінансових втрат;
- Неправильна інформація про наявність товарів. Відсутність точної інформації про товари, може спричинити до неправильних замовлень, які

своєю чергою несуть магазину репутаційну втрату. Неактуальна або неповна інформація про наявність товарів може бути результатом ручного обліку, помилок у внесених даних або відсутності інформаційної системи;

- Погане планування. Недостатнє планування про потреби та постачання є ще однією важливою проблемою. Недооцінка або переоцінка потреб призведе до наслідків з товарами. Погане планування може створити проблеми з неправильним використанням ресурсів, затримки з постачаннями;
- Повільна реакція на зміни потреб клієнтів. Магазины повинні бути наготові для реагування на зміни потреб та забезпечити швидку появу необхідних товарів у заданий проміжок. Повільна реакція на зміни попиту може призвести до збитків від продажу, конкурентоспроможності та втрату клієнтів.

Процес замовлення та постачання товарів є одним з багатьох елементів управління будівельними магазинами. Неясність в цих процесах може призвести до низки проблем, які в перспективі значно ускладнюють роботу магазину і впливають на подальшу роботу з клієнтами. Проблеми та їх наслідки:

- Затримки в постачанні. Відсутність інформаційної системи для поставок та слабка комунікація з постачальниками можуть призвести до проблем у постачанні товарів. Несвоєчасне доставлення матеріалів та інструментів до магазину може призвести до перебоїв замовлень клієнтів, втрати довіри та незадоволення з їх боку.
- Неправильний облік товарів. Така ситуація виникає, коли система показує, що товар є на складі, але його немає. Також можуть виникнути випадки, коли товари є на складі, але система вважає їх відсутніми.
- Помилки в комунікації: Ручне введення даних та обмін інформацією замовлення з постачальниками можуть призвести до помилок в комунікації.

Управління клієнтами є важливою діяльністю будівельних магазинів. Приклади проблем, які можуть виникати в цьому напрямку, необхідно детально розглянути та зосередитися на пошуку ефективних рішень.

- Додавання, збереження та оновлення клієнтських даних. Однією з основних проблем є збереження та оновлення клієнтської інформації. Зростаюча тенденція в базі клієнтів і потреба у збереженні детальних даних про них (контактна інформація, ПІБ тощо) може стати складною задачею без використання відповідної інформаційної системи. Відсутність призведе до втрати часу на пошук в ручному вигляді клієнтів з детальною інформацією. Рішення відбувається в тому, щоб створити інформаційну систему, яка дозволить зберегти та оновити інформацію про ваших клієнтів точно та швидко. Це забезпечить продуктивність будівельного магазину.
- Швидке обслуговування. Для забезпечення високого рівня обслуговування клієнтів необхідно розробити детальні процедури та інструкції, особливо якщо працівники не мають достатнього досвіду або працюють вперше. Рішення: одним із рішень для забезпечення швидкого обслуговування є використання торгових касових апаратів для ефективної обробки платежів. Крім того, рішенням є інформаційна система підтримки будівельного магазину, яка оптимізує процеси обслуговування. Для співробітників така практика дозволяє вдосконалювати свої навички обслуговування, системного адміністрування та спілкування з клієнтами. Для покращення інтеграції потрібна інформаційна система, яка об'єднувала різні аспекти діяльності магазину, такі як керування товарами, продажами та фінансовими операціями. Це дозволить автоматизувати багато процесів та забезпечити швидку обробку даних.

Фінансовий облік є одним з важливих складових в управлінні будівельними магазинами. Фінансові операції дають змогу забезпечити стабільність, контроль і прибутковість підприємства. Без автоматизації фінансового обліку та фінансових операцій це може призвести до багатьох

проблем, які важко вирішити, а також може негативно вплинути на фінансову стабільність і продуктивність самого магазину.

Виявлені проблеми, що виникають при управлінні будівельними магазинами, свідчать про необхідність використання сучасних інформаційних систем в даній сфері. Тому необхідність використання сучасних технологій є виправдано для покращення керування цими підприємствами. Автоматизовані системи управління товарами, продажами та обслуговуванням клієнтів, а також фінансові операції можуть в суттєвій мірі спростити процеси та підвищити ефективність роботи будівельних магазинів.

1.2. Аналіз інформаційних систем аналогів

Сфера керування будівельними магазинами має потребу у використанні інформаційних систем, які в результаті дають оптимізацію процесів, забезпечення швидкості обробки даних та підвищити ефективність роботи магазину в цілому.

Спеціалізоване програмне забезпечення також варто розглянути, як інформаційна система підтримки роботи будівельного магазину. Такі системи забезпечують широкий функціонал, що можуть робити різні операції з товарами, клієнтами, контроль товарів та створювати звітність. Також використовуються для швидкої обробки даних та ефективного керування процесами будівельного магазину. Однак, підтримка таких систем можуть бути затратними, якщо магазин має невеликий бюджет. Для використання такого програмного забезпечення необхідні спеціальні знання та навички, що може стати додатковим викликом для персоналу магазину.

«BAS Малий бізнес» є програмним забезпеченням(Рис.1.1), розробленим компанією Ukrainian Intelligent Technology, яке призначено для управління товарами та продажами в будівельних магазинах. Це програмне забезпечення надає широкий функціонал і має на меті полегшити різні процеси діяльності магазину.

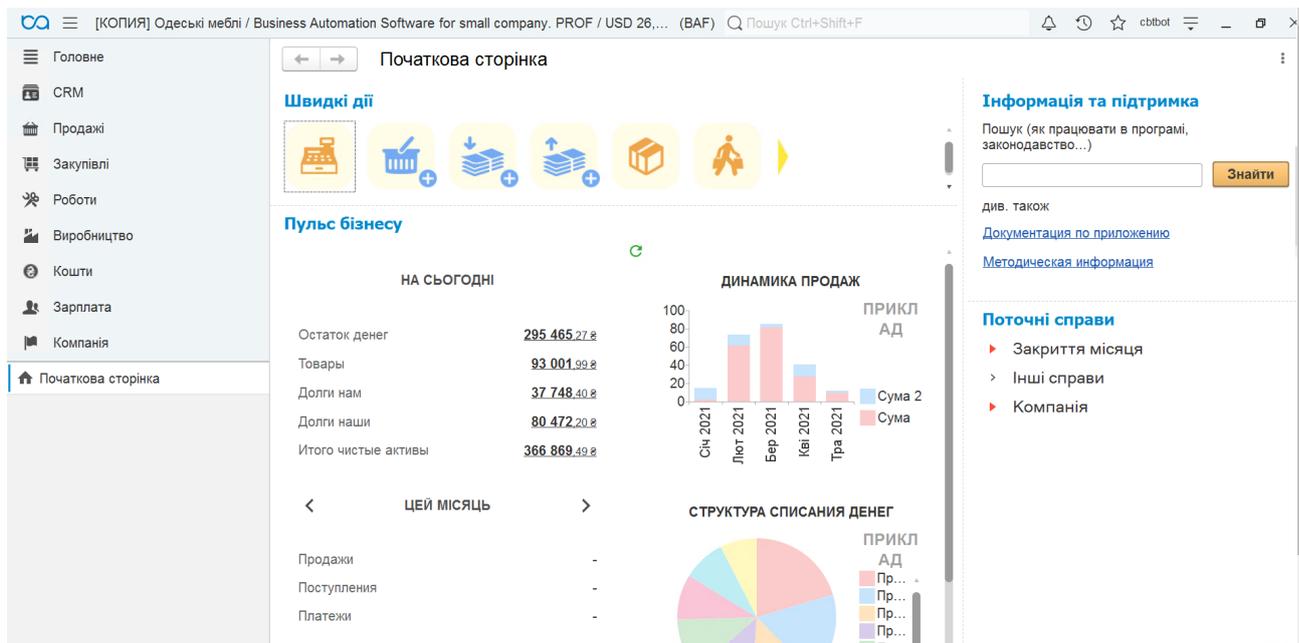


Рис.1.1. Головна форма «BAS Малий бізнес»

Проведемо оцінку відповідності вимогам "BAS Малий бізнес" для будівельного магазину, а також спробуємо визначити переваги, недоліки та можливості, що інформаційна система може надати. Розглянемо загальні вимоги для будівельного магазину:

- Функціональні вимоги: чи задовольняє «BAS Малий бізнес» основні потреби магазину щодо управління товарами та продажами, включаючи облік клієнтів, звітність і фінансові операції?
- Технічні вимоги: чи дозволяє сумісності програмного забезпечення з наявною інфраструктурою магазину інтеграції з іншими системами (наприклад, касова система, термінал для оплати картою)?

Основні можливості «BAS Малий бізнес» включають наступне:

- Управління процесами товарів: система дозволяє вести детальний облік товарів, організацію процесу закупівлі, накладна, що оформляє операції надходження товарів від постачальників і повернення товарів від покупців, здійснювати продаж та отримувати повідомлення про недостачу товару. Останнє допомагає уникнути недостатньої кількості товару[2-3].

- Управління продажами: розділ «Гроші» програми BAS Малий бізнес містить документи для проведення розрахунків у готівковій та безготівковій формі, планування надходження та вибуття грошей, аналізу доступності коштів на кожен день. Документ «Чек ККМ» використовується для реєстрації продажу та оплати товарів. Також можливе створення звітів та аналіз продажів для прийняття виробничих рішень(Рис.1.2)[2-3].

Рис.1.2. Форма для продажу в «BAS Малий бізнес»

- Касове обслуговування: система має вбудований касовий модуль для зручного розрахунку з покупцями, приймання платежів і фінансових операцій(Рис.1.3)[2-3].

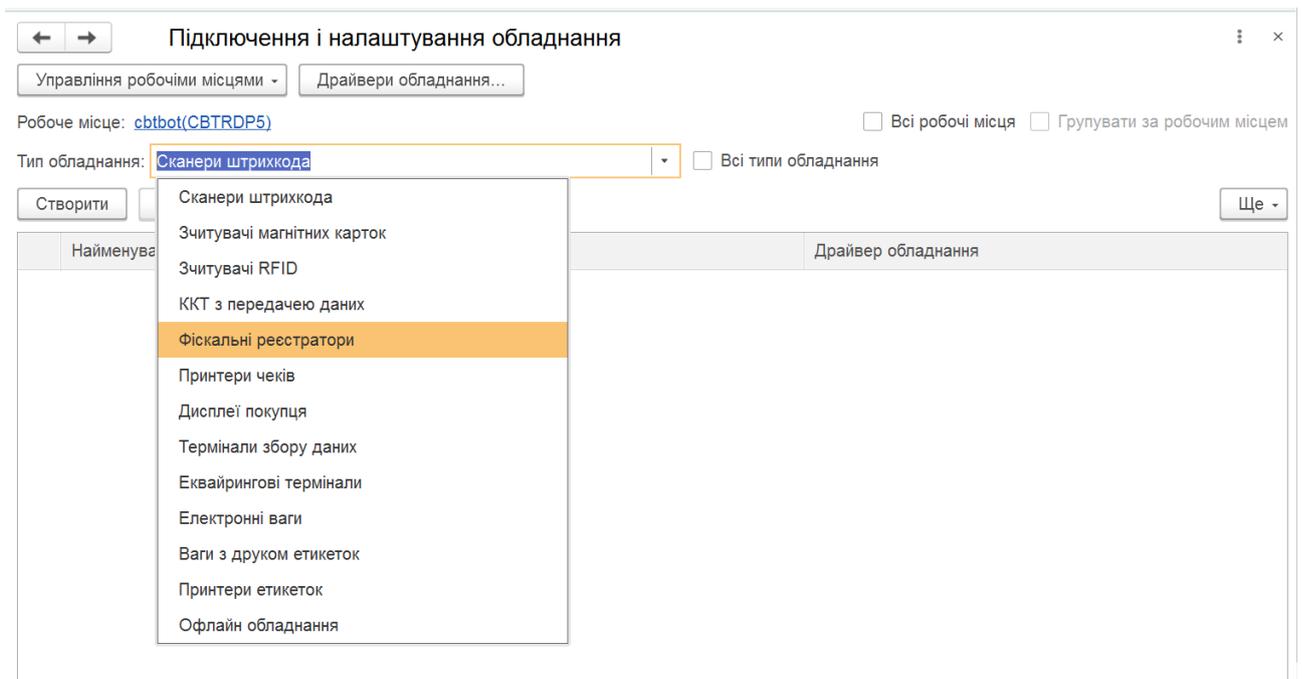


Рис.1.3. Форма для підключення в «BAS Малий бізнес»

- Клієнтська база даних: завдяки системі можна виконувати різні операції з базою даних клієнтів[2-3].
- Звітність і аналітика: система надає можливість згенерувати різноманітні звіти про продажі, товари, прибуток. Ці звіти допомагають зробити обдумані рішення для підвищення прибутковості та популярності магазину(Рис.1.4)[2-3].

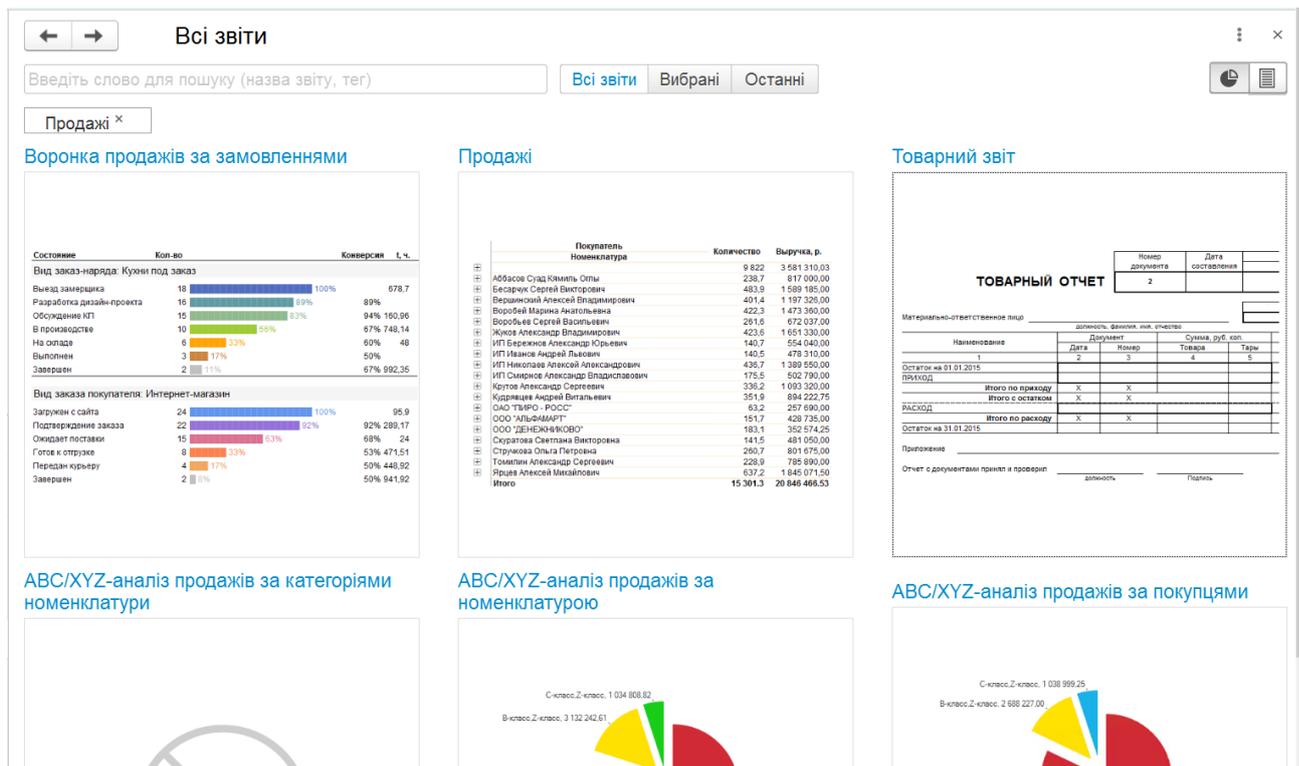


Рис.1.4. Звіти та аналітика даних «BAS Малий бізнес»

Інформаційна система "BAS Малий бізнес" відповідає вимогам будівельного магазину, надаючи повний контроль над товарами, продажами та фінансовими операціями. Ця система спеціально розроблена для використання в невеликих магазинах. Під час дослідження було виявлено як переваги, так і вади цієї інформаційної системи. Переваги «BAS Малий бізнес» включають:

- Автоматизація процесів: система дозволяє замінити ручну роботу на автоматизовані процеси, що зменшує кількість помилок та покращує продуктивність[2-3].
- Швидкість та точність: завдяки функціям та зрозумілому інтерфейсу, система забезпечує швидкий доступ до інформації[2-3].
- Збереження часу і зусиль: автоматизовані функції дозволяють прискорити процеси, такі як створення звітів та обробка продажів, зменшуючи час і зусилля, які б були необхідними для виконання цих завдань[2-3].

- Покращений контроль: система надає детальну інформацію про стан товарів, продажів і фінансових операцій, що дозволяє бути в курсі ситуації та при необхідності приймати рішення[2-3].

Недоліки «BAS Малий бізнес» включають[2-3]:

- Було виявлено, що витрати на впровадження та підтримку цього програмного забезпечення можуть бути значними для малого бізнесу.
- Не підтримує ведення обліку операцій більш як однієї фірми на одній інформаційній базі.
- Тільки один користувач може одночасно використовувати базу даних.
- Впровадження системи вимагає певного рівня технічних навичок.
- В залежності від інфраструктури, можуть виникати проблеми, якщо підприємство не здатне підтримувати програмне забезпечення.
- не підтримується зміна конфігурації, можна використовувати лише типову конфігурацію та встановлювати її оновлення.

Переходимо до ефективності впровадження «BAS Малий бізнес»:

- Вартість програмного забезпечення: вартість ліцензії на програмне забезпечення для одного користувача становить 504 грн. Це досить доступна вартість, особливо для малих бізнесів. Якщо необхідно декілька ліцензій, вартість буде відповідно збільшуватися. Наприклад, якщо будівельний магазин має 10 користувачів, загальна вартість ліцензії становитиме 5 040 грн на рік.
- Вартість впровадження: вартість впровадження системи «BAS Малий бізнес» становить від 4 200 грн. Ця сума включає витрати на впровадження, а також навчання персоналу. Вартість може змінюватися в залежності від обсягу робіт та складності інтеграції з іншими системами(наприклад терміналами для оплати картою).
- Вигода: якщо подивитися на функціональні можливості «BAS Малий бізнес», можна сказати, що є деякі конкретні функції, які б хотіли отримати від впровадження. Наприклад, ви можете використовувати

реєстрацію для продажу та оплати товарів, звіти з продажу дозволять проаналізувати стан виконання операцій з клієнтами. Як для інформаційної системи підтримки невелика ціна вартості ліцензії та встановлення на необхідну інфраструктуру. Ці вигоди можуть призвести до збільшення продуктивності та покращення звітності.

- Час для освоєння інформаційної системи підтримки персоналом: Нова система може вимагати певного часу для навчання персоналу. Наприклад, ознайомлення з базою даних "UNIPRO" був успішно пройдений протягом одного тижня.

Програмне забезпечення для керування операціями з товарами та продажами є незамінним інструментом для будівельних магазинів. Воно допомагає автоматизувати процеси керування товарами, замовленнями та продажами, що дозволяє підвищити функціональність роботи магазину.

Отже, інформаційна система «BAS Малий бізнес» має недоліки такі, як вартість системи, складність обробки великого обсягу даних, складна у навчанні, потребує відповідну інфраструктуру, тому було вирішено розробити просту інформаційну систему, яка має мінімальні функціональні можливості, які забезпечать роботу магазину. Розроблена інформаційна система є вартою розгляду для магазинів, які шукають функціонал для керування товарами та продажами.

РОЗДІЛ 2. Опис моделі та методів її реалізації.

2.1. Вимоги до інформаційної системи

Сучасні технології не обійшли стороною будівельну галузь, а саме будівельні матеріали. Після війни вся країна буде відбудовувати зруйновані будинки, дороги, будувати нові будівлі. Тут потрібні будівельні магазини, які стануть перевальним пунктом для матеріалів. Щоб не втратити продуктивність в будівельному магазині, необхідно просто використовувати інформаційну систему підтримки роботи магазину.

В інформаційній системі підтримки роботи будівельного магазину необхідно мати можливість запису обліку та реєстрації в системі. Він складається з наступних пунктів:

- Головна;
- Клієнти;
- Каталог товарів;
- Каса;
- Поставки;
- Інформація про персонал.

Користувач повинен мати інформацію про співробітників магазину. Обов'язково врахувати, щоб користувач зміг додавати до бази нові поставки (накладні), товари, клієнти. Редагування та видалення теж входить до обробки даних. При роботі з операціями по обробці даних повинні проводитися деякі перевірки на правильність введених даних. При вибраному рядку з інформацією про постачання повинно надрукувати накладну. Пункт «Каса» повинний означати оплату вибраного товару. Каса складається з таких пунктів:

- Пошук товару;
- Додавання знижки;
- Керування та зберігання товару;
- Звіти про продажі.

Для роботи з даними було створено базу даних за допомогою Microsoft Access. Сама система створена за допомогою мови С#. Розробка графічного інтерфейсу інформаційної системи підтримки роботи будівельного магазину створена з використанням додатків Windows Forms.

Метою розробки інформаційної системи підтримки роботи будівельного магазину є продукт, який вирішить проблему автоматизації управління будівельним магазином, помилки при ручному складанні даних, збереже час персоналу завдяки електронному обігу даних.

2.2. Опис моделі інформаційної системи

Будівельні магазини мають здебільшого однаковий організаційний склад. Нижче показано просту організаційну діаграму(Рис.2.1).

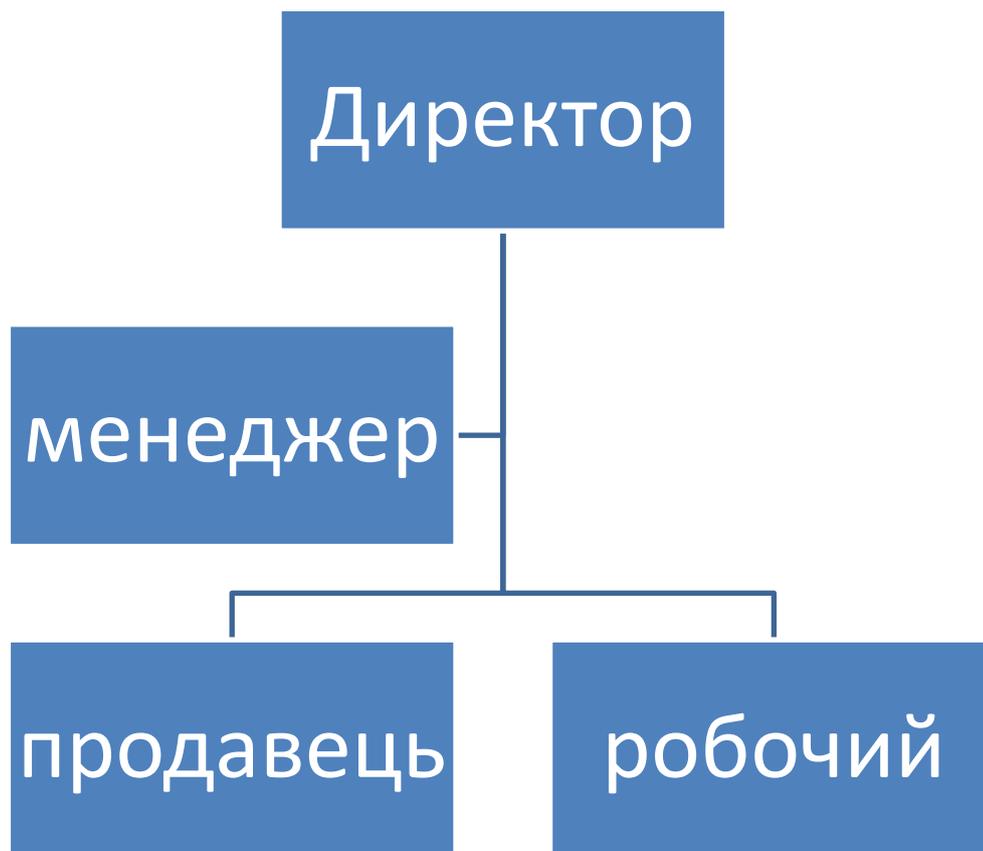


Рис.2.1. Організаційна діаграма

Далі розглядається встановлення функціональних вимог, необхідних для розробки інформаційної системи, що підтримує діяльність будівельного магазину:

- Можливість додавання інформації про новий товар.
- Функціонал для перегляду інформації про співробітників даного магазину.
- Можливість редагувати та видаляти інформацію про товар.
- Функціонал для поставок тобто накладних, що прибули.
- Друк для накладних і пошук за датою і заданими ключовими словами.
- Інформація про постачальників.
- Функціонал для продажу товару.
- Можливість додавання знижки до товарів.
- Функціонал для звітності про щоденні продажі.
- Збереження інформації про оплату.
- Авторизація користувача.

Представлена загальна структура інформаційної системи, яка підтримує роботу будівельного магазину. Робота інформаційної системи, яка підтримує різні операції в будівельному магазині, потребує входу в систему, даних авторизації користувача або реєстрації, запиту користувача на додавання нових продуктів, запиту на продаж продуктів, створення звітів про продажі або додавання знижок, запит на збереження платежів і редагування продуктів, доставки, клієнти, інформація про персонал. Виконує управління інформаційною системою користувач. На рисунку 2.2. показано структурну схему інформаційної системи, що розробляється.



Рис.2.2. Загальна схема системи

Для більш повної концепції бізнес-логіки системи показано на рисунку 2.3 діаграма класів, що дає зрозуміти функції та хто виконує.

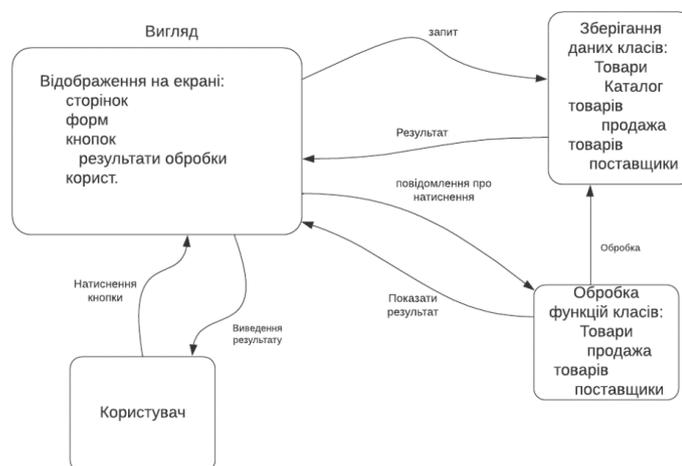


Рис.2.3. Архітектура системи

У таблиці «Вигляд» відображається інтерфейс системи: форма з товарами та інформацією про них, форми для входу і реєстрації в системі, кнопки та інші результати обробки. Колонка з назвою «Зберігання даних» чітко вказує, для чого вона створена.

Табличка з назвою «Обробка функцій» обробляє дані для таблиці, перелічених у таблиці «Зберігання даних». Крім того, таблиця «Користувач» показує, що один користувач виконує роботу.

2.3. Проектування системи за допомогою UML-діаграм

Для розуміння логіки побудови інформаційної системи було розроблено декілька діаграм UML. Зокрема, були використані наступні діаграми: діаграма класів, компонентів, прецедентів, активностей та послідовностей[4-6].

UML, яка є уніфікованою мовою моделювання, використовувалась для візуалізації систем програмного забезпечення та керування процесами. Наприклад, діаграми UML демонструють бізнес-процеси та надають детальну інформацію для розуміння системи і його перетворення у програмний код.

На малюнку 2.4 показана діаграма класів, яка містить класи, методи та зв'язки між ними.

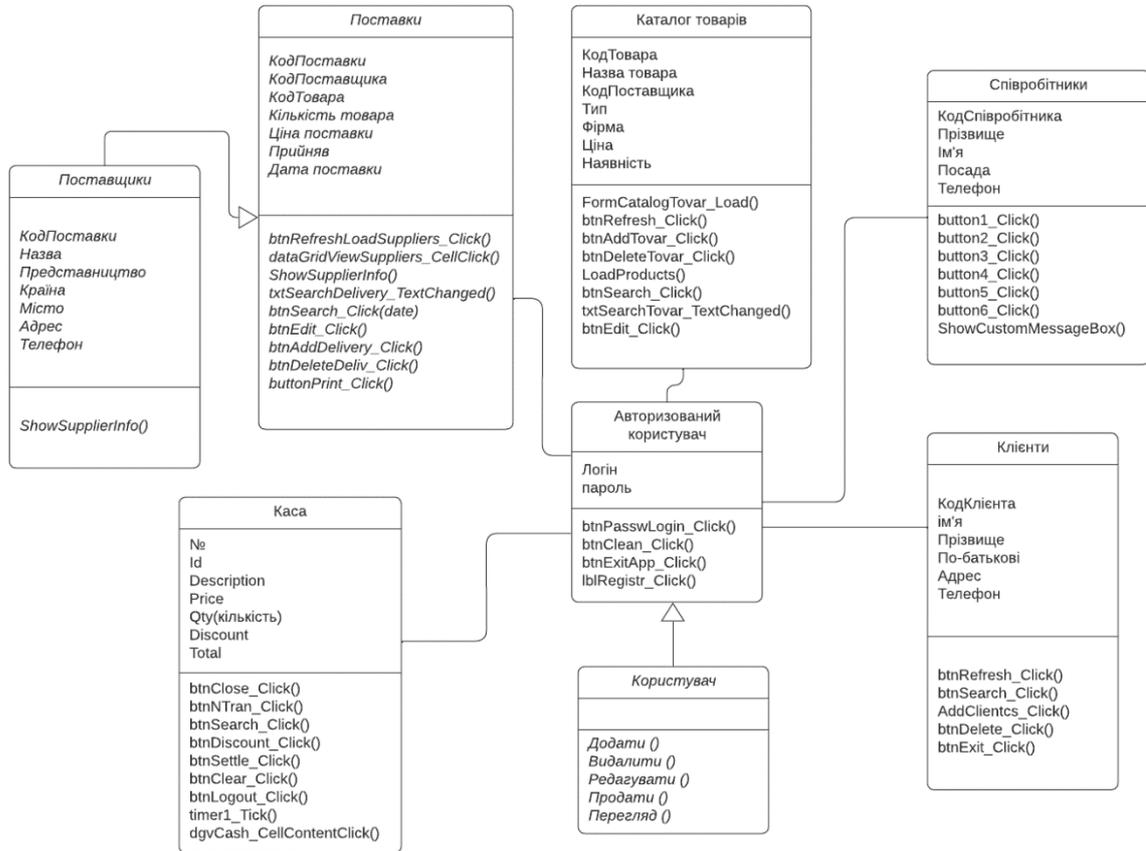


Рис.2.4. Діаграма класів системи

Наступна діаграма ілюструє частини інформаційної системи, які складають систему. Діаграма компонентів більш читабельна, ніж діаграма класів, тому що один компонент реалізується не тільки одним класом, а й кількома. Тобто, якщо система яка розробляється є великою, то діаграма компонентів є незамінним чинником, яка показує роботу та взаємодію між собою у великому масштабі для досягнення певного функціонала.

Діаграма компонентів для інформаційної системи підтримки будівельного магазину зображена на рисунку 2.5.

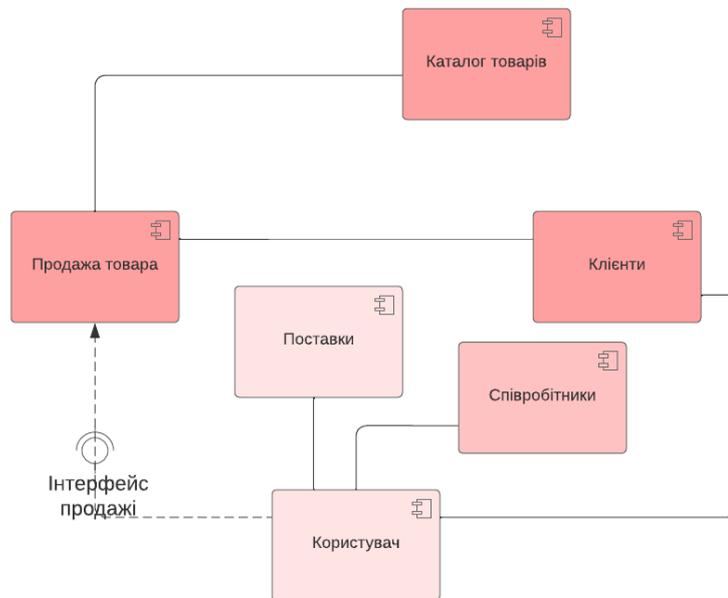


Рис.2.5. Діаграма компонентів системи

Діаграма компонентів використовується з метою наглядного представлення функціоналу системи з точки зору користувача, пояснення роботи цього функціоналу або формування вимог до системи. Вона складається з наступних компонентів: актор, який представляє особу або підсистему, що взаємодіє з системою; прецедент, який відповідає на запитання про те, що система робить у певній ситуації, не розкриваючи, як саме це відбувається; система, яка представляє інформаційну систему підтримки будівельного магазину; та зв'язок, який відображає лінію комунікації між актором і прецедентом.

Наступне зображення показує вхід у систему підтримки будівельного магазину(Рис.2.6).

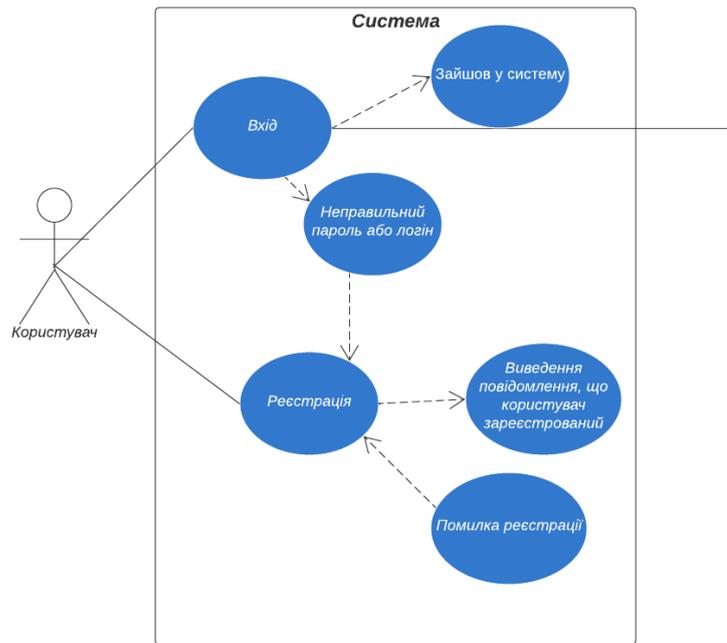


Рис.2.6. Діаграма прецедент «Реєстрація в системі»

У роботі розглядаються два варіанти подій, пов'язаних з процесом авторизації в інформаційній системі. Перший варіант — користувач вводить логін і пароль і натискає кнопку «Вхід», після чого відкривається головне вікно системи. Другий варіант передбачає ситуацію, коли після введення даних виникає помилка «Неправильний логін та пароль», і користувач переходить на вікно реєстрації. Якщо в цьому вікні є порожні поля, які потрібно заповнити, з'являється повідомлення про помилку «Поля ім'я користувача та пароль є пустими». У разі, якщо всі дані введені коректно, з'являється повідомлення про успішну реєстрацію.

Наступний прецедент який називається «Каса» зображений на рисунку 2.7.

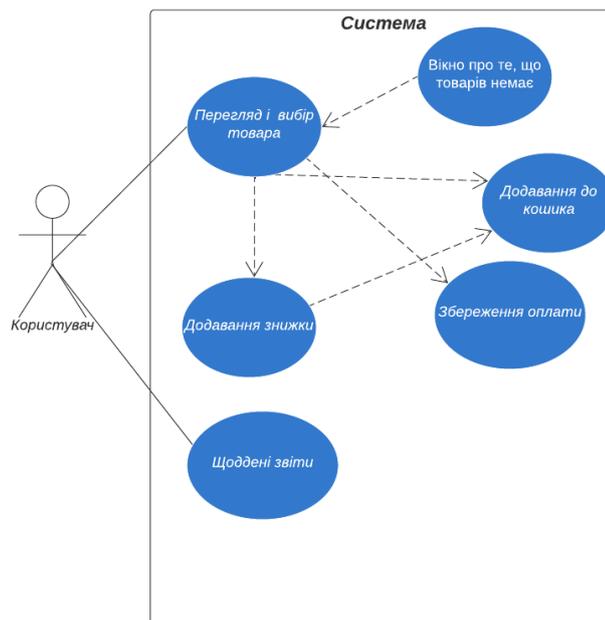


Рис.2.7. Діаграма прецедент «Каса»

В роботі у вкладці «Каса» реалізовано наступний функціонал:

- 1) Користувач здійснює вхід у вкладку «Каса».
- 2) Користувач може скористатись кнопкою «Щоденні звіти», яка відображає продажі товарів за певний період.
- 3) Користувач може виконати пошук товару для його вибору. Якщо товару немає, з'являється повідомлення про помилку «Товарів немає». Після цього він має можливість скористатись опціями, такими як застосування знижки або збереження оплати.

Діаграма прецедентів «Каталог товарів» має схожий інтерфейс з кнопками, такими як «Поставки», але додатково містить опції, наприклад, пошук за датою та інформацію про постачальника. Діаграма показана на рисунку 2.8.

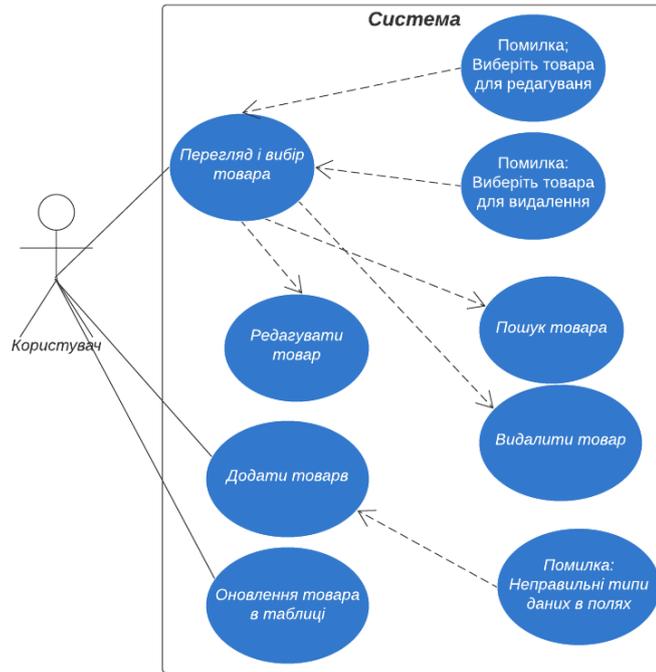


Рис.2.8. Діаграма прецедент «Каталог товару»

Діаграма для пошуку поставки в системі зображена на рисунку 2.9.

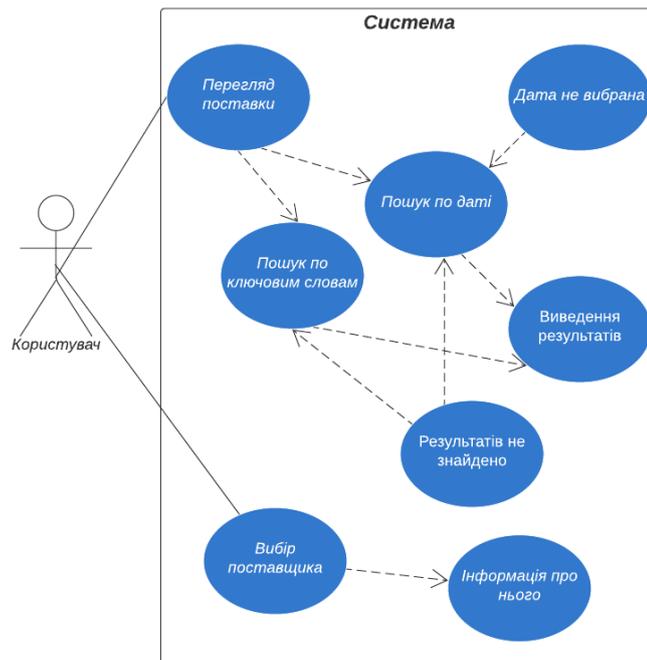


Рис.2.9. Діаграма прецедент «Пошук поставки»

Робота з пошуком містить наступні етапи:

1) Користувач переходить до вкладки «Поставки» і обирає потрібний варіант пошуку. Якщо результатів не знайдено, система виводить повідомлення про помилку «Результатів не знайдено». Крім того, якщо користувач не вибере дату для пошуку матеріалів, з'явиться повідомлення про помилку «Дату не вибрано».

2) Користувач вибирає постачальника, після чого система відображає інформацію про цього постачальника.

Була використана діаграма активностей, яка надає наочне зображення поведінки системи, тобто процесу, як одна функція переходить до іншої. Наведено нижче діаграму активностей (Рис.2.10).

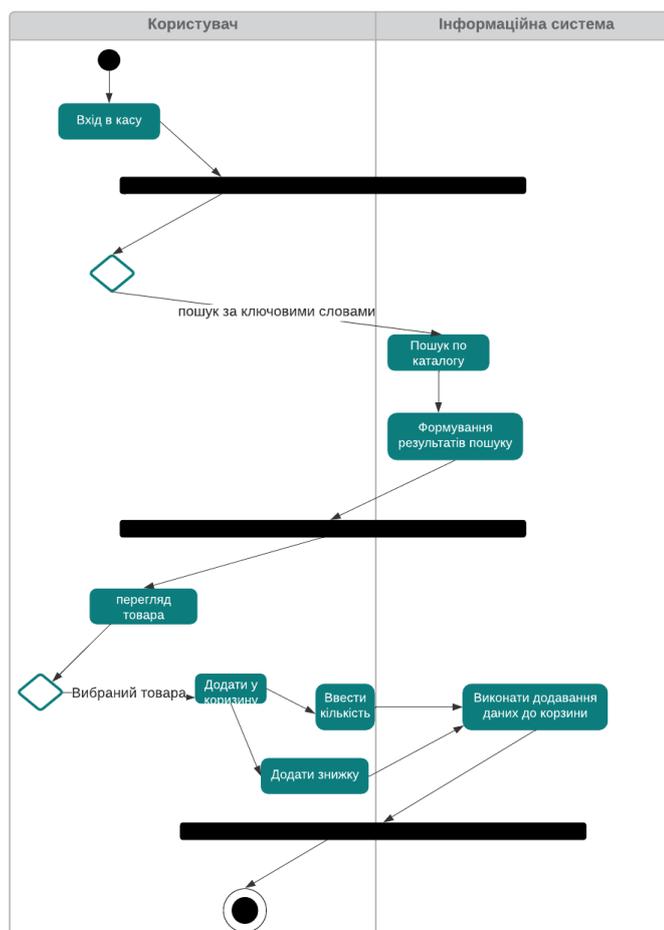


Рис.2.10. Діаграма активностей продажу товару

Наступний малюнок, який знаходиться нижче(Рис.2.11 – 2.12) показує реєстрацію та вхід у інформаційній системі.

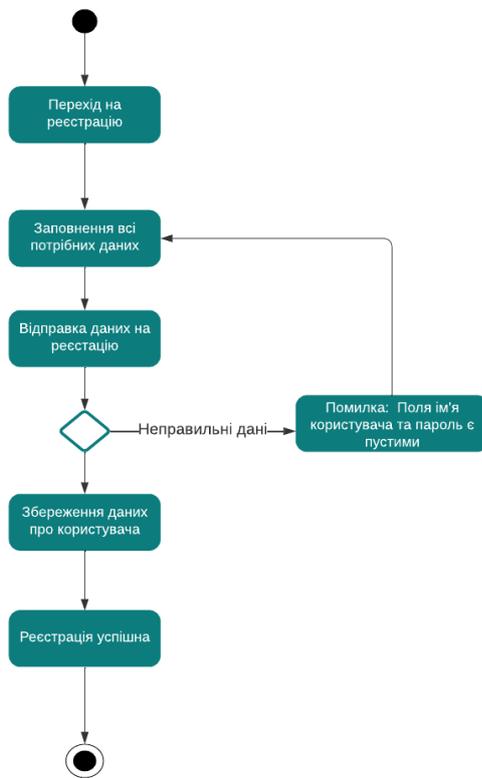


Рис.2.11. Діаграма активностей реєстрації

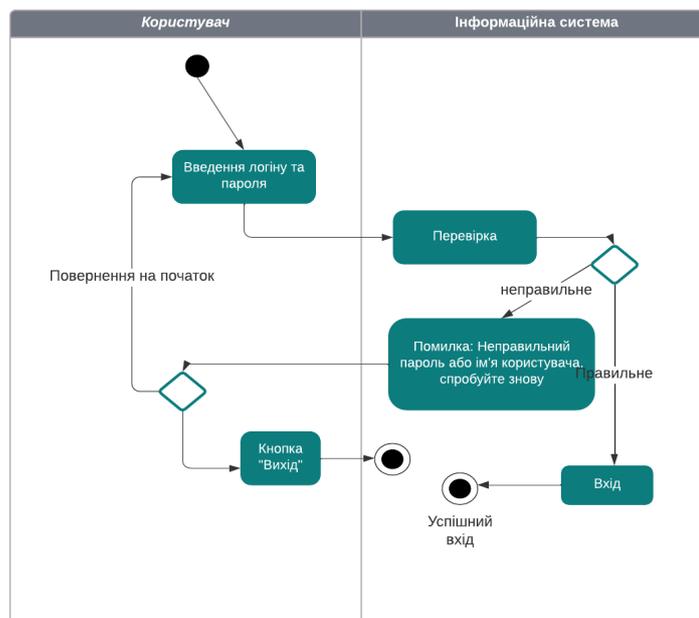


Рис.2.12. Діаграма активностей входу

Діаграма послідовностей показує чітку роботу між процесами під час виконання. Нижче зображено на рисунку 2.13 пошук товару.

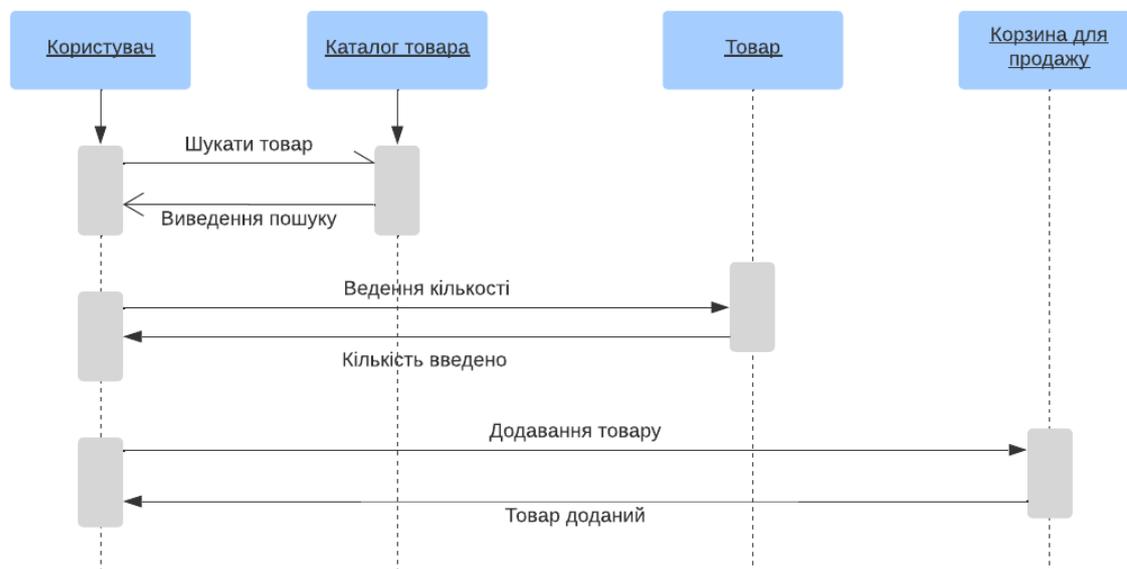


Рис.2.13. Діаграма послідовності пошуку товару

2.4. Вибір технологій розробки

Є багато факторів, які визначає людина при розробці проекту. Наприклад, одними з таких факторів є безпека та зручність використання технологій у роботі з проектом. У контексті спілкування з комп'ютером, ми використовуємо різні мови програмування, що нагадує розмову між людьми. Як засоби спілкування ці мови програмування мають свої переваги та недоліки, які суб'єктивно можна виключити. Основною вимогою є те, щоб людина, що працює у сфері програмування, розуміла обрану мову програмування. Навіть якщо деякі мови програмування є популярними протягом багатьох років, враховуючи тенденції у технологічному світі, все може швидко змінитися. На рисунку 2.14 представлена статистика щодо популярності та використання різних мов програмування в сучасному часі.

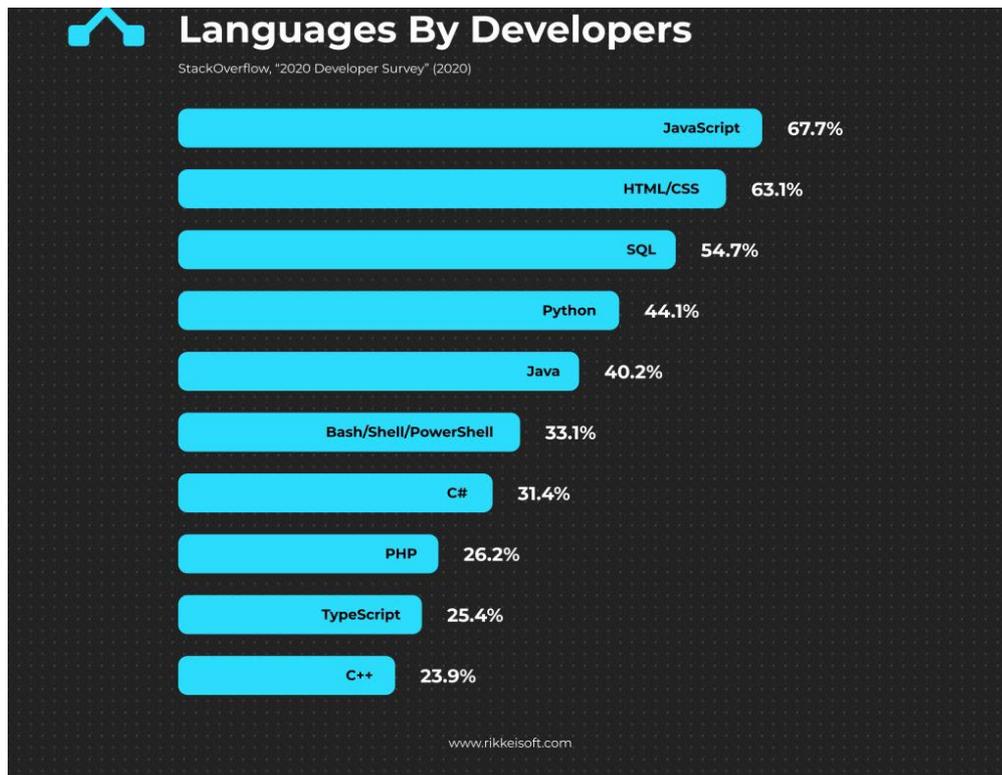


Рис.2.13. Статистика популярних мов

Перед вибором мови програмування для інформаційної системи підтримки будівельного магазину, проаналізуємо декілька відомих мов.

JavaScript – правитель інтерфейсу. По статистиці вона була найбільш популярна у 2022. Мова дуже проста у вивченні завдяки своїй простоті. Проста людина може вивчити її за пару місяців і самостійно створити додаток чи інтерфейс[7-10].

HTML/CSS – це дві мови, які створюють можливість красивих та функціональних веб-сторінок. Ці мови можна побачити на будь-якому сайті в браузері. Проста мова HTML, яка дозволяє створити розмітку, а CSS доповнює її для оформлення веб-сторінок[7-10].

Python – одна з найбільш популярних мов. Такі додатки, як Instagram, Spotify, Youtube та багато інших були створені нею. Надає різні можливості для розробки ігор, веб-сайтів тощо[7-10].

Java – теж популярна і доступна для різноманітних розробок. В ній є деякі особливості такі, як простота логіки, об'єктно-орієнтовне програмування

(допомагає створити більш структуровану програму, використовуючи об'єкти), безпека, робота на різних платформах(Windows, macOS тощо)[7-10].

C# - це об'єктно-орієнтована мова програмування, як відповідь Microsoft на Java. Використовується для програм Windows, мобільних додатків та інших[7-10].

PHP – призначена для загального використання в розробці веб-додатків та веб-сторінок. Вона легка для розуміння синтаксису, багато різних бібліотек та фреймворків, підтримує базу даних[7-10].

Тепер порівняємо декілька мов програмування за критеріями(табл.2.1).

Таблиця 2.1 – Порівняння мов

Критерії	Python	Java	C#	PHP
Тип	Інтерпретована	Компільована	Компільована	Інтерпретована
Синтаксис	Простий	Суворий, об'єктно-орієнтований	об'єктно-орієнтований	процедурний і об'єктно-орієнтований
Залежність від платф.	Повна	Велика	Залежить від реалізації .NET Framework	Повна
Продуктивність	Середня	Висока	Висока	Середня
Веб-розробка	вбудовані фреймворки	вбудовані фреймворки	вбудовані фреймворки	вбудовані фреймворки

З аналізу попередніх даних випливає, що для розробки інформаційної системи підтримки будівельного магазину, є вигідним використання мови програмування C# та технології Windows Forms. Використання цих інструментів забезпечує сумісність з платформою, широкий набір

функціональних можливостей, наявність різноманітних бібліотек та простоту використання.

В основі інформаційної системи лежить ідея, яка забезпечує зберігання різноманітної інформації про покупки, персонал та інші дані. Для досягнення цієї мети необхідно використовувати базу даних, щоб відповідати ефективному зберіганню та організації цих даних.

База даних використовується в багатьох середовищах. Вона також надає інструменти для звіту та інші результати на основі даних. Існує кілька типів баз даних[11-12]:

- Реляційні бази даних. Зручні для простого користування, які дають можливість для зберігання даних в таблиці. Приклади програм: MySQL, Oracle, SQLite, Access.
- Об'єктно-орієнтовані. Складніша база даних для керування складними зв'язками даними. Дані зберігаються у формі об'єкта. Приклади програм: Couchbase, Neo4j, фреймворк Hibernate.
- Так звані бази даних NoSQL. Призначення яких є зберігання та керування не реляційними даними. Приклади програм: MongoDB, Redis, Apache Cassandra.

Проаналізуємо реляційні бази даних такі, як MySQL та Access.

MySQL є незамінною системою зберігання та управління даними, яка займає ключову позицію в галузі IT-технологій. MySQL широко використовується Facebook, Twitter, YouTube та багатьма іншими організаціями. Створена у 1995 році, нині MySQL продовжує бути відкритою системою, що сприяє вільному використанню та зміні її коду. База даних є популярним вибором серед розробників веб-додатків, веб-сайтів та інших проектів завдяки її перевагам ефективного та надійного зберігання даних[11-12].

Програма Access, що входить до пакета Microsoft, є невід'ємною складовою для збереження та маніпулювання даними. Через свою простоту база даних стає популярним вибором для багатьох компаній, магазинів та

інших організацій. Використовуючи Access, можна створювати таблиці для зберігання даних, звіти для виведення основних відомостей, запити для здійснення пошуку та інші функції, а також форми, які надають зручний інтерфейс для додавання інформації[11-12]. Порівняємо ці дві бази даних(табл.2.2).

Таблиця 2.2 – Порівняння баз даних

Критерії	MySQL	Access
Тип бази даних	Реляційна	Реляційна
Підтримка платформи	Різні системи	Тільки Windows
Можливості	Відкритий код, масштабований, безпечний, швидкість та надійність	Зручний та зрозумілий інтерфейс, шаблони, конструктор запитів, інтеграція з іншими програмами
Складність використання та навчання	Додаткове налаштування та складне навчання	Просте використання та навчання

З порівняння бази даних можна побачити, що база даних MySQL має переваги в ефективному та надійному зберіганні даних, а також у масштабованості та безпеці. Однак, використання MySQL може вимагати додаткового налаштування та складного навчання. У зв'язку з цим, де обсяг даних у будівельному магазині є мінімальним, прийнято рішення використовувати базу даних Microsoft Access. Вона володіє простим та зрозумілим інтерфейсом, має шаблони та конструктори запитів, а також інтеграцію з іншими програмами. Такий вибір дозволить легко керувати базою даних та забезпечить можливість самостійного виправлення користувачем. Хоча база даних Microsoft Access має обмежену підтримку платформи та

можливостей порівняно з MySQL, вона забезпечує достатній рівень функціональності та зручності в управлінні даними.

РОЗДІЛ 3. Опис програмної реалізації інформаційної системи.

3.1. Створення бази даних

В базі даних Access створено десять таблиць з необхідними назвами полів. Нижче на рисунку 3.1 зображено першу таблицю під назвою «Товари».

	Имя поля	Тип данных
	КодТовара	Счетчик
№	Назва товара	Короткий текст
	КодПоставщика	Короткий текст
	КодТипа	Короткий текст
	КодФірми	Короткий текст
	ЦінаПродажу	Денежный
	Наявність	Логический

Рис.3.1. Таблица товарів

Поле «КодПоставщика» має зв'язок з таблицею «Поставщики». Він показує назву постачальника товару. Реалізовано на SQL-запиті або за допомогою Майстера підстановок. Код для SQL-запита:

```
SELECT [Поставщики].[Назва] FROM Поставщики ORDER BY [Назва];
```

Та ж сама ситуація і для полів «КодТипа» і «КодФірми». Код SQL-запит для типу і фірми:

```
SELECT [Типи].[Категорія] FROM Типи ORDER BY [Категорія];
```

```
SELECT [Назва фірми].[Марка] FROM [Назва фірми];
```

Наступна таблиця показує дані про типи товарів, які існують. Маленька табличка з одним рядком(Рис.3.2).

	Имя поля	Тип данных
№	Категорія	Короткий текст

Рис.3.2. Таблица типів товару

Таблиця «Співробітники» створена для інформації про персонал(Рис.3.3).

Имя поля	Тип данных
КодСпівробітника	Счетчик
Прізвище	Короткий текст
Ім'я	Короткий текст
Посада	Короткий текст
Телефон	Короткий текст

Рис.3.3. Таблиця співробітників

Наступний рисунок 3.4 показує інформацію про постачальників.

Имя поля	Тип данных
КодПоставщика	Счетчик
Назва	Короткий текст
Представництво	Короткий текст
Країна	Короткий текст
Місто	Короткий текст
Адрес	Короткий текст
Телефон	Короткий текст

Рис.3.4. Таблиця постачальників

Таблиця «Поставка» показує поставки під відповідними полями(Рис.3.5)

Имя поля	Тип данных
КодПоставки	Счетчик
КодПоставщика	Короткий текст
КодТовара	Короткий текст
КількістьТовара	Числовой
ЦінаПоставки	Денежный
Прийняв	Короткий текст
Дата поставки	Дата и время

Рис.3.5. Таблиця поставок

В поставках ситуація схожа, тільки появилoся запит для товару та співробітника. SQL-запити:

SELECT [Товари].[Назва товара] **FROM** Товари **ORDER BY** [Назва товара];

SELECT [Співробітники].[Прізвище] **FROM** Співробітники **ORDER BY** [Прізвище];

Таблиця «Назва фірми» теж має одне поле під назвою «Марка». Зображено на рисунку 3.6.

Имя поля	Тип данных
Марка	Короткий текст

Рис.3.6. Таблица фірми

Таблиця «Клієнти» показує інформацію, яка написана в полях таблиці(Рис.3.7).

Имя поля	Тип данных
КодКлієнта	Счетчик
Ім'я	Короткий текст
Прізвище	Короткий текст
По батькові	Короткий текст
Адрес	Короткий текст
Телефон	Короткий текст

Рис.3.7. Таблица клієнтів

Для запам'ятовування пароля та логін, щоб увійти в інформаційну систему, створено таблицю «Profile_Users», яка зображена на рисунку 3.8.

Имя поля	Тип данных
Username	Короткий текст
Password	Короткий текст

Рис.3.8. Таблица зберігання даних для входу

Для пошуку товарів і додавання їх до кошика створюється таблиця «Products», як показано на рисунку 3.9.

	Имя поля	Тип данных
PK	ID_Product	Счетчик
	PCode	Короткий текст
	Barcode	Числовой
	Description	Короткий текст
	Brand	Короткий текст
	Category	Короткий текст
	Price	Денежный
	Qty	Числовой

Рис.3.9. Таблица для пошуку і додавання товару

Поле під назвою «Qty» означає кількість продукту на складі. Також є зв'язок між таблицями «Товари», «Назва фірми» та «Типи».

Остання таблиця під назвою «Cart» означає кошик для продажу вибраного товару і збереження даних проданих товарів. Нижче показано таблицю на рисунку 3.10.

	Имя поля	Тип данных
PK	ID_Cart	Счетчик
	Transno	Короткий текст
	Pcode	Короткий текст
	Price	Числовой
	Qty	Числовой
	Disc_percent	Числовой
	Description	Короткий текст
	Disc	Числовой
	total	Числовой
	sdate	Дата и время
	status	Короткий текст
	cashier	Короткий текст

Рис.3.10. Таблица «Корзина»

В кінцевому результаті виходи така схема даних, зображена на рисунку 3.11.


```

        // Отримати інформацію про постачальника за допомогою
supplierId
        using (OleDbConnection connection = new
OleDbConnection(ConnectionString))
        {
            string query = "SELECT * FROM Поставщики WHERE
[Назва] = @SupplierNazva;";
            using (OleDbCommand command = new
OleDbCommand(query, connection))
            {
command.Parameters.AddWithValue("@SupplierNazva", supplierId);
                connection.Open();

                using (OleDbDataReader reader =
command.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        // Отримати дані про постачальника
                        string supplierName =
reader.GetString(reader.GetOrdinal("Назва"));
                        string supplierPredstav =
reader.GetString(reader.GetOrdinal("Представництво"));
                        string supplierCountry =
reader.GetString(reader.GetOrdinal("Країна"));
                        string supplierCity =
reader.GetString(reader.GetOrdinal("Місто"));
                        string supplierAddress =
reader.GetString(reader.GetOrdinal("Адрес"));
                        string supplierPhoneNum =
reader.GetString(reader.GetOrdinal("Телефон"));

                        // Відображення інформації про
постачальника

```



```

        // Формування запиту до бази даних
        string query = "SELECT * FROM Поставка WHERE
КодПоставщика LIKE @Keyword OR Прийняв LIKE @Keyword OR КодТовара
LIKE @Keyword;";

        OleDbCommand command = new OleDbCommand(query,
connection);

        command.Parameters.AddWithValue("@Keyword",
"%"+ searchKeyword + "%");

        // Виконання запиту до бази даних та отримання
результатів

        OleDbDataAdapter adapter = new
OleDbDataAdapter(command);

        DataTable dataTable = new DataTable();
        adapter.Fill(dataTable);

        // Відображення результатів у DataGridView
        dataGridViewSuppliers.DataSource = dataTable;

    }
    catch (Exception ex)
    {
        // Обробка винятку при з'єднанні з базою даних
        MessageBox.Show("Помилка при з'єднанні з базою
даних: " + ex.Message, "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}
}
}

```

Код запити:

```

SELECT * FROM Поставка WHERE КодПоставщика LIKE @Keyword OR
Прийняв LIKE @Keyword OR КодТовара LIKE @Keyword

```

Він вибирає всі рядки з таблиці «Поставка», де поле «КодПоставщика», «Прийняв» або «КодТовара» містять певний шаблон, заданий параметром @Keyword.

3.3. Опис інтерфейсу системи

Для реалізації інформаційної системи підтримки будівельного магазину використано простий та сучасний інтерфейс, що включає такі компоненти:

- TextBox.
- Button.
- ListBox.
- Label.
- Panel.

Усі ці компоненти включені у Windows Forms, тому додаткове завантаження фреймворків чи бібліотек не потрібне. На рисунку 3.12 показана перша форма з відповідним інтерфейсом, а на рисунку 3.13 - друга форма.

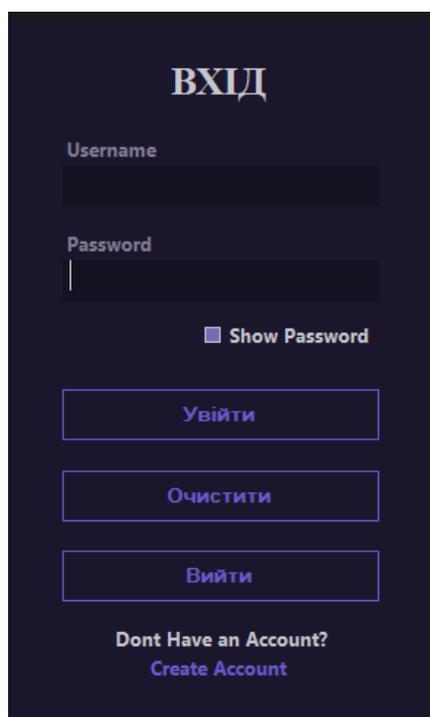


Рис.3.12. Інтерфейс входу

РЕЄСТРАЦІЯ

Username
[Input field]

Password
[Input field]

Confirm Password
[Input field]

Show Password

[Зареєструватися](#)

[Очистити](#)

Already Have an Account?
[Back to LOGIN](#)

Рис.3.13. Інтерфейс реєстрації

Головна форма після входу у систему показана на рисунку 3.14, а також інтерфейс для клієнтів на рисунку 3.15.

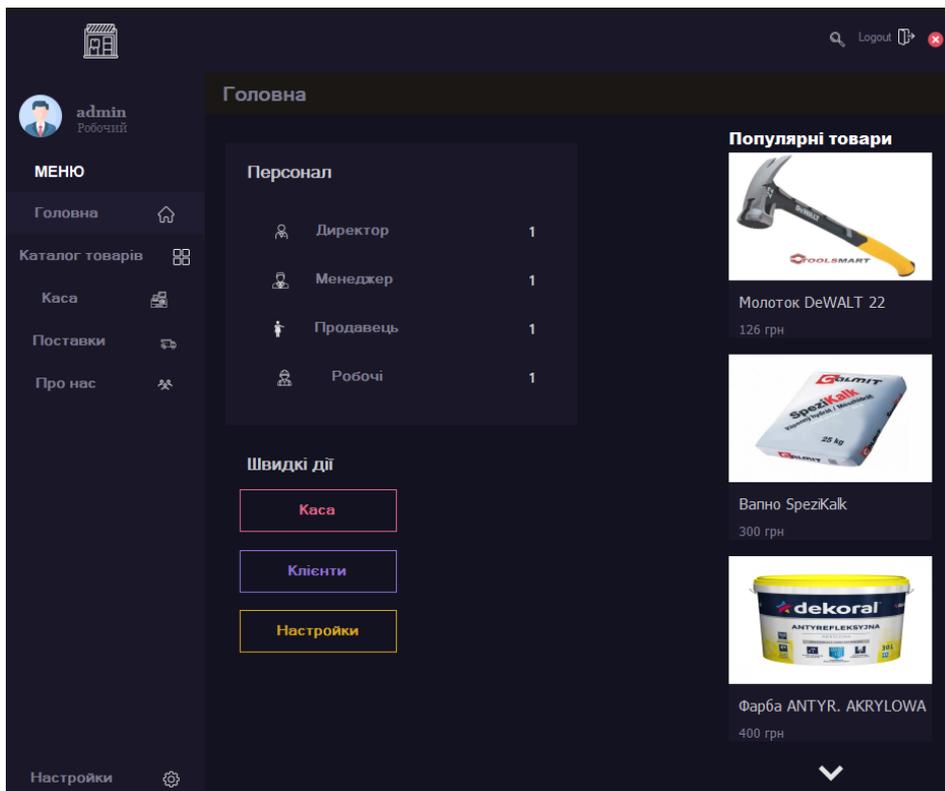


Рис.3.14. Інтерфейс головної форми

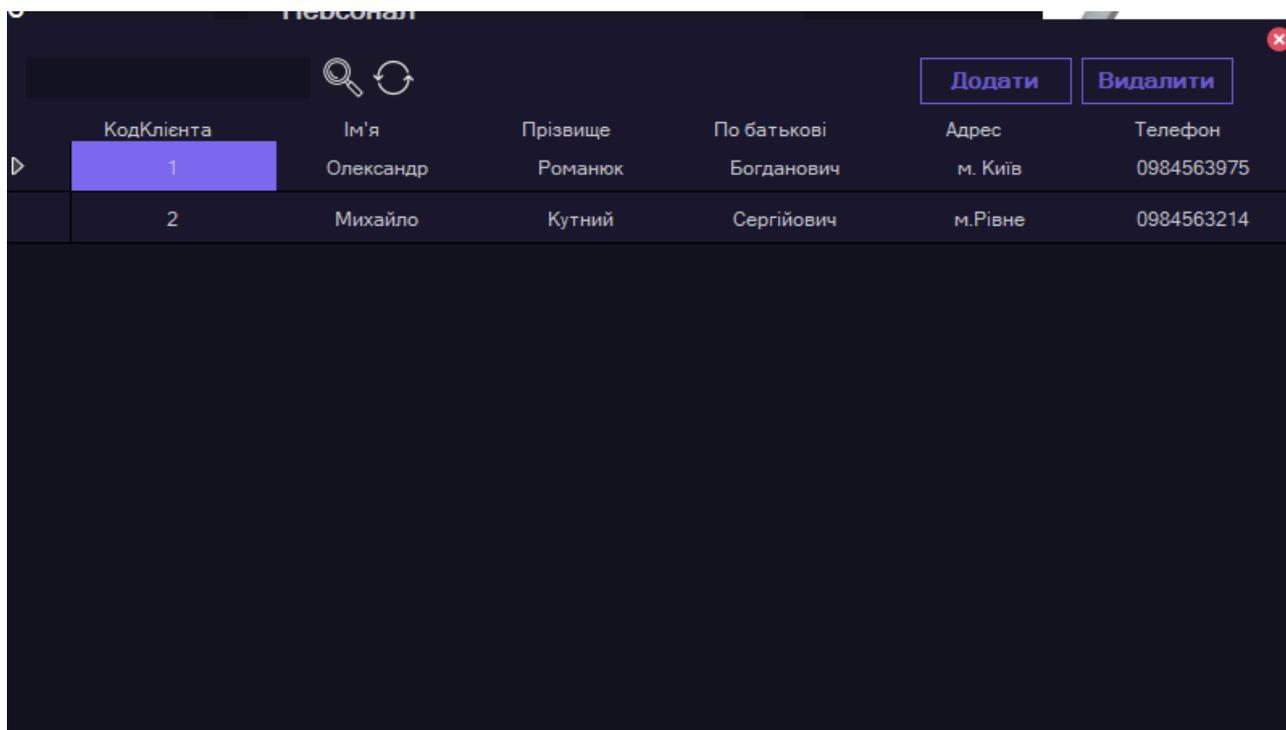


Рис.3.15. Інтерфейс клієнтів

Інтерфейс додавання клієнта появляється при активації кнопки «Додати». Зображений на рисунку 3.16.

Ім'я:

Прізвище:

По-батькові:

Адрес:

Телефон:

Рис.3.16. Інтерфейс додавання клієнта

У всіх наступних інтерфейс додавання буде схожий на цей, крім полів, які міняються в залежності вибраної кнопки. Наступний інтерфейс зображено на рисунку 3.17 та 3.18.

Каталог товарів

admin
Робочий

МЕНЮ

Головна

Каталог товарів

Каса

Поставки

Про нас

Настройки

КодТовара	Назва товара	КодПоставщика	КодТипа	КодФірми	ЦінаПродажу	Наявність
2	Цемент	Інструмент	Сировинні	virgips	90	<input checked="" type="checkbox"/>
3	Деревноволо...	CoDLACzeba	Готові до зас...	BRYZA	1000	<input type="checkbox"/>
4	Фанера	GrupTools	Напівафабрик...	PENOBBOARD	234	<input checked="" type="checkbox"/>
5	Цегла	ZTF grup	Готові до зас...	RakaMakafo	10	<input checked="" type="checkbox"/>
7	Керамічна пл...	CoDLACzeba	Напівафабрик...	Dyckerhoff	110	<input checked="" type="checkbox"/>
8	Паркетна кле...	Сrona+	Сировинні	Jukhfm	75	<input checked="" type="checkbox"/>

Рис.3.17. Інтерфейс для товарів

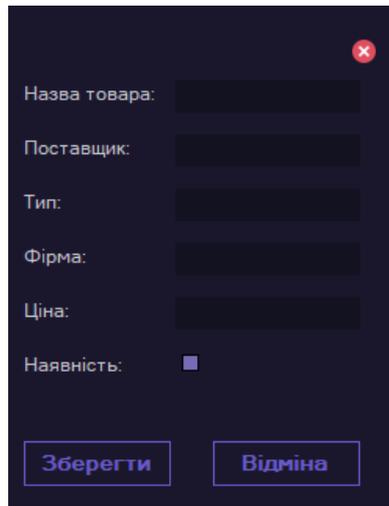


Рис.3.18. Інтерфейс для редагування товару

Інтерфейс для співробітників зображений на рисунку 3.19.

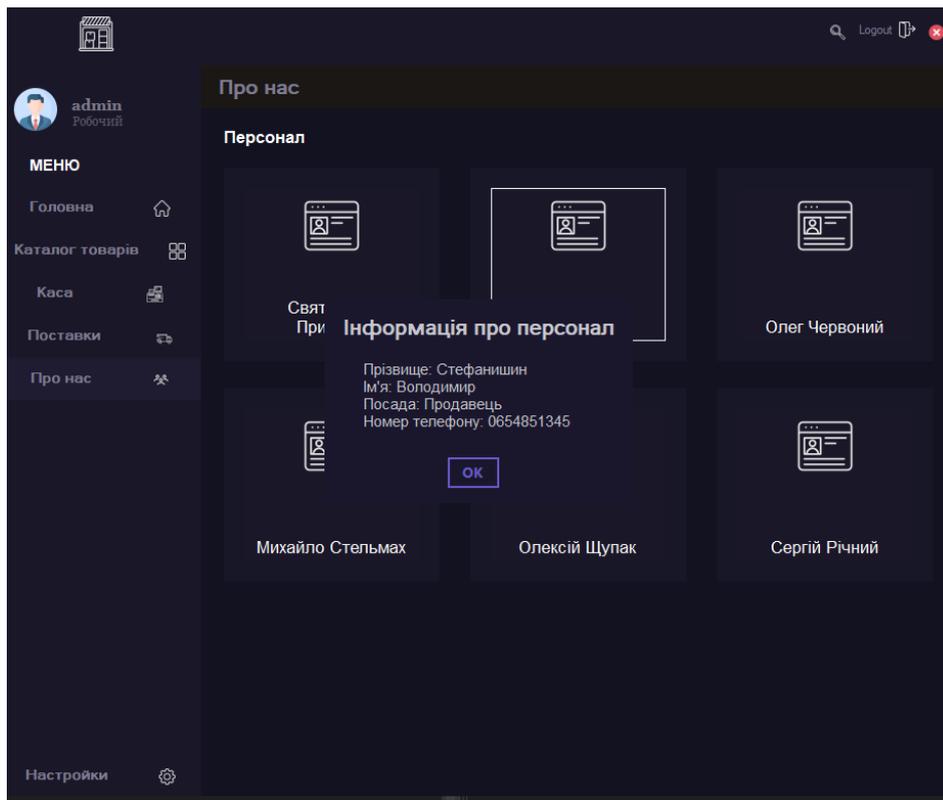


Рис.3.18. Інтерфейс про персонал

Інтерфейс для поставок зображений на рисунку 3.19, де є можливість для друку.

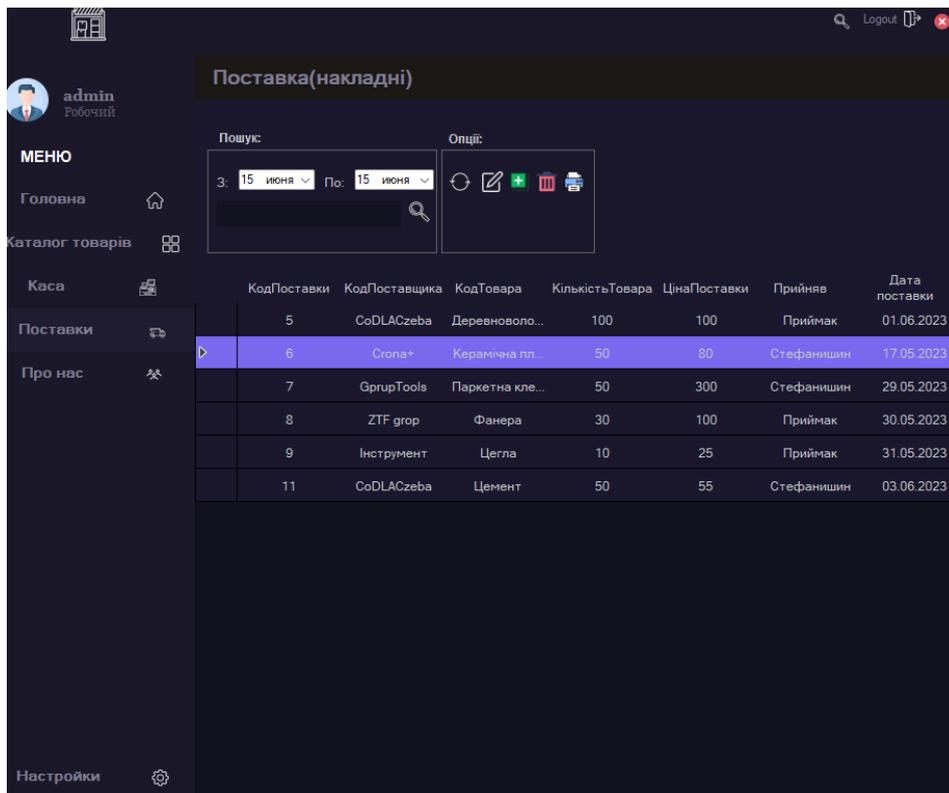


Рис.3.19. Інтерфейс для поставок

Натиснувши кнопку "Каса", з'являється інтерфейс, який призначений для здійснення продажу товарів. Форма для продажу товарів зображена на рисунку 3.20, а на рисунках 3.21-3.23 показано інтерфейс для додавання знижок, пошуку товару для додавання в кошик, керування оплатою та отримання щоденних звітів по продажам.

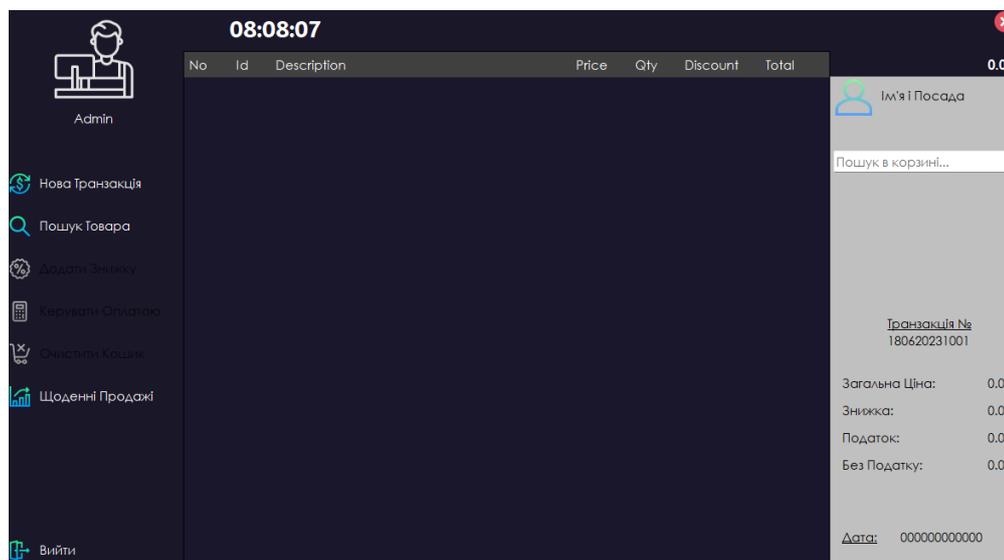


Рис.3.20. Інтерфейс для продажу товару

No	Pcode	Barcode	Description	Brand	Category	Price	Qty
1	P001	654321	Керамічн...	BRYZA	Готові до застосування	100	5
2	P002	654322	Паркетна ...	Dyserhoff	Напівфабрикати	250	15
3	P003	654323	Деревнов...	RELAX	Готові до застосування	500	30
4	P004	654324	Фанера	PENOBORD	Готові до застосування	50	47
5	P005	654325	Цегла	РакТА	Готові до застосування	10	150
6	P006	654326	Цемент	Jukhfm	Сировинні	100	25

Пошук Товара:

Закрити

Рис.3.21. Інтерфейс для пошуку товару і додавання в корзину

Знижка ✕

Загальна Ціна: 100,00

Знижка:

Сума Знижки:

Підтвердити

Рис.3.22. Інтерфейс додавання знижки

Керування Оплатою ✕

100,00

7	8	9	C
4	5	6	0
1	2	3	00

Enter

Рис.3.23. Інтерфейс для керування оплатою

3.4. Опис застосунка

Інформаційна система будівельного магазину для управління магазином, створена за допомогою C# .NET Windows Forms та бази даних Access.

Створена система надає можливості, додати товар, пошук товару, продати товар та інші додаткові функції, видаляти, друкувати, пошук по датам поставки(накладні), пошук клієнтів, перегляд інформації та оновлення її.

Зареєструватися можуть працівники підприємства, магазину та інші. Якщо користувач не зареєстрований, то при нажаті кнопки «Create Account» відкривається інтерфейс в якому потрібно ввести логін та пароль. При успішній реєстрації працівник нажимає на кнопку «Back to LOGIN» і вводить логін та пароль.

Для вибору товарів для покупки створена кошик, де користувач може виконати продаж. Система має зручний та простий інтерфейс у використанні.

ВИСНОВКИ

Сучасні інформаційні системи мають значний вплив на численні сфери людської діяльності. Вони мають значний вплив на бізнес, уряд, освіту, медицину, виробництво та інші сфери. Впровадження інформаційних систем дозволяє уникнути проблем, пов'язаних з паперовою роботою, а також зменшити ризик помилок, які можуть призвести до втрати репутації та прибутковості.

Для ефективного функціонування будівельного магазину, необхідна швидка та точна обробка різних функцій, таких як продаж товару, приймання поставок тощо.

Була розроблена інформаційна система підтримки будівельного магазину, яка забезпечує збереження інформації, можливість перегляду, функціонал для продажу товару покупцям, різні опції для обробки даних та звітності по продажах. Інтерфейс системи простий і зрозумілий, що дозволяє користувачам швидко освоїти його за обмежений проміжок часу.

Для розробки інформаційної системи була вибрана мова програмування C#, оскільки вона надає безкоштовний доступ до багатьох бібліотек, різноманітний функціонал, а також можливість створення інтерфейсу за допомогою фреймворк Windows Forms.

Для забезпечення оптимального використання інформаційної системи в будівельному магазині, який може мати обмежений обсяг та ресурси, можна використовувати реляційну базу даних Access. Недоліки, які вона має перебиваються її простотою, швидкістю навчання та легкістю керування.

Для покращення інформаційної системи можна розробити базу даних на основі MySQL, оскільки вона має більші можливості у роботі з даними та підходить для великого обсягу інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. EEMERITUS. What is Information System? Definition, Examples, & Facts. URL: <https://emeritus.org/in/learn/information-system/> (дата звернення 19.05.2023)
2. Ukrainian Intelligent Technology. BAS Малий бізнес. URL: <https://uit.kiev.ua/product/bas-malij-biznes/> (дата звернення 22.05.2023)
3. Rentsoft cloud technologies. BAS Малий бизнес (BAS small business). URL: <https://rentsoft.ua/ru/catalog/bas-malij-biznes-prof/> (дата звернення 22.05.2023)
4. Miro. What are UML diagrams, and how can you use them? URL: <https://miro.com/blog/uml-diagram/> (дата звернення 01.06.2023)
5. Visual Paradigm. What is Unified Modeling Language (UML)? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> (дата звернення 01.06.2023)
6. IBM. Creating use-case diagrams. URL: <https://www.ibm.com/docs/en/rsar/9.5?topic=diagrams-creating-use-case> (дата звернення 02.06.2023)
7. FreeCodeCamp. What is Computer Programming? URL: <https://www.freecodecamp.org/news/what-is-programming/> (дата звернення 07.06.2023)
8. HP TECH. Computer History: A Timeline of Computer Programming Languages. URL: <https://www.hp.com/us-en/shop/tech-takes/computer-history-programming-languages> (дата звернення 07.06.2023)
9. NEXTTECHNOLOGY. The most popular programming languages in 2023. URL: <https://nexttechnology.io/the-most-popular-programming-languages-in-2023/> (дата звернення 10.06.2023)
10. Berkeley Extension. 11 Most In-Demand Programming Languages. URL: <https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/> (дата звернення 11.06.2023)

11. Simplilearn. What is a Database? Everything You Need to Know. URL: <https://www.simplilearn.com/tutorials/dbms-tutorial/what-is-a-database> (дата звернення 12.06.2023)
12. Talend. What is MySQL? Everything You Need to Know. URL: <https://www.talend.com/resources/what-is-mysql/> (дата звернення 12.06.2023)
13. Microsoft. OleDbConnection Клас. URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbconnection?view=dotnet-plat-ext-7.0> (дата звернення 13.06.2023)
14. Гужва В.М. Інформаційні системи і технології на підприємствах: Навчальний посібник / В.М. Гужва – Київ: КНЕУ, 2001. — 400 с.
15. Карпенко М. Ю. Конспект лекцій з курсу «Інформаційні системи і технології в управлінні організацією. Частина 1» / М. Ю. Карпенко, В. Б. Уфимцева; Харк. нац. акад. міськ. госп-ва; – Х.: ХНАМГ, 2012. – 96 с.
16. Плескач В.Л. Інформаційні системи і технології на підприємствах: підручник / В.Л. Плескач, Т.Г. Затонацька. – К. : Знання, 2011. – 718 с.
17. Томашевський О. М., Цегелик Г. Г., Вітер М. Б., Дудук В. І. Інформаційні технології та моделювання бізнес-процесів: навч. посіб. для студентів вищ. навч. закл. / О. М. Томашевський [та ін.]. - Київ : Центр учбової літератури, 2012. - 296 с.

ДОДАТОК А

FormDelivery

```
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.Data.OleDb;
using System.Drawing.Printing;

namespace BYDMIX
{
    public partial class FormDelivery : Form
    {

        public string ConnectionString =
            "Provider=Microsoft.ACE.OLEDB.12.0;" +
            "Data Source= D:\\Visual
studio\\Bakalavrskaja_robota_BYDMIX_2023\\BYDMIX\\BYDMIX\\DIY_store_
db.accdb";

        public FormDelivery()
        {
            InitializeComponent();
        }

        private void btnRefreshLoadSuppliers_Click(object sender,
EventArgs e)
        {
```

```

        using (OleDbConnection connection = new
OleDbConnection(ConnectionString))
        {
            try
            {
                connection.Open();
                string query = "SELECT * FROM Поставка";
                OleDbDataAdapter adapter = new
OleDbDataAdapter(query, connection);
                DataTable dataTable = new DataTable();
                adapter.Fill(dataTable);
                dataGridViewSuppliers.DataSource = dataTable;
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка підключення до бази
даних: " + ex.Message);
            }
        }

        private void dataGridViewSuppliers_CellClick(object
sender, DataGridViewCellEventArgs e)
        {
            // Перевірити, чи було натиснуто на ячейку стовпця
"КодПостачальника"
            if (e.ColumnIndex ==
dataGridViewSuppliers.Columns["КодПоставщика"].Index && e.RowIndex
>= 0)
            {
                // Отримати значення КодПостачальника з вибраної
ячейки
                string supplierIdValue =
dataGridViewSuppliers.Rows[e.RowIndex].Cells["КодПоставщика"].Value
e.ToString();

```

```

        // Викликати функцію відображення інформації про
постачальника з передачею supplierId
        ShowSupplierInfo (supplierIdValue);
    }
}

private void ShowSupplierInfo (string supplierId)
{
    // Отримати інформацію про постачальника за допомогою
supplierId
    using (OleDbConnection connection = new
OleDbConnection (ConnectionString))
    {
        string query = "SELECT * FROM Поставщики WHERE
[Назва] = @SupplierNazva;";
        using (OleDbCommand command = new
OleDbCommand (query, connection))
        {
            command.Parameters.AddWithValue ("@SupplierNazva", supplierId);
            connection.Open ();

            using (OleDbDataReader reader =
command.ExecuteReader ())
            {
                if (reader.Read ())
                {
                    // Отримати дані про постачальника
                    string supplierName =
reader.GetString (reader.GetOrdinal ("Назва"));
                    string supplierPredstav =
reader.GetString (reader.GetOrdinal ("Представництво"));
                    string supplierCountry =
reader.GetString (reader.GetOrdinal ("Країна"));
                }
            }
        }
    }
}

```

```

        string supplierCity =
reader.GetString(reader.GetOrdinal("Місто"));
        string supplierAddress =
reader.GetString(reader.GetOrdinal("Адрес"));
        string supplierPhoneNum =
reader.GetString(reader.GetOrdinal("Телефон"));

        // Відображення інформації про
постачальника

        MessageBox.Show($"Назва:
{supplierName}\nПредставництво: {supplierPredstav}\nКраїна:
{supplierCountry}\nМісто: {supplierCity}\nАдрес:
{supplierAddress}\nТелефон: {supplierPhoneNum}", "Інформація про
постачальника");
    }
    else
    {
        MessageBox.Show("Постачальника не
знайдено", "Помилка");
    }
}
}
}

private void txtSearchDelivery_TextChanged(object sender,
EventArgs e)
{
    string searchKeyword = txtSearchDelivery.Text;

    // Підключення до бази даних
    using (OleDbConnection connection = new
OleDbConnection(ConnectionString))
    {

```

```

try
{
    // Відкриття з'єднання з базою даних
    connection.Open();

    // Формування запиту до бази даних
    string query = "SELECT * FROM Поставка WHERE
КодПоставщика LIKE @Keyword OR Прийняв LIKE @Keyword OR КодТовара
LIKE @Keyword;";

    OleDbCommand command = new OleDbCommand(query,
connection);

    command.Parameters.AddWithValue("@Keyword",
"% " + searchKeyword + "%");

    // Виконання запиту до бази даних та отримання
результатів

    OleDbDataAdapter adapter = new
OleDbDataAdapter(command);

    DataTable dataTable = new DataTable();
    adapter.Fill(dataTable);

    // Відображення результатів у DataGridView
    dataGridViewSuppliers.DataSource = dataTable;

}
catch (Exception ex)
{
    // Обробка винятку при з'єднанні з базою даних
    MessageBox.Show("Помилка при з'єднанні з базою
даних: " + ex.Message, "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
}
}

```

```

private void listBoxResults_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (listBoxResults.SelectedIndex != -1)
    {
        string selectedResult =
listBoxResults.SelectedItem.ToString();
        txtSearchDelivery.Text = selectedResult;
        listBoxResults.Visible = false;

        // Оновіть DataGridView з результатами пошуку,
використовуючи вибране ключове слово
        LoadProducts();
    }
}

private void LoadProducts()
{
    using (OleDbConnection connection = new
OleDbConnection(ConnectionString))
    {
        try
        {
            connection.Open();
            string query = "SELECT * FROM Поставка";
            OleDbDataAdapter adapter = new
OleDbDataAdapter(query, connection);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);
            dataGridViewSuppliers.DataSource = dataTable;
        }
        catch (Exception ex)
        {

```

```

        MessageBox.Show("Помилка підключення до бази
даних: " + ex.Message);
    }
}

private void btnSearch_Click(object sender, EventArgs e)
//пошук по датам
{
    // Отримати введені дати для пошуку
    DateTime fromDate = dateTimePickerFrom.Value.Date;
    DateTime toDate = dateTimePickerTo.Value.Date;

    // Пройтися по кожному рядку DataGridView і знайти ті,
що відповідають введеним датам
    foreach (DataGridViewRow row in
dataGridViewSuppliers.Rows)
    {
        // Отримати значення дати з відповідної комірки
        DateTime date =
Convert.ToDateTime(row.Cells[6].Value);

        // Перевірити, чи дата потрапляє в діапазон
        if (date >= fromDate && date <= toDate)
        {
            // Відобразити рядок, якщо він відповідає
умовам пошуку
            row.Visible = true;
        }
        else
        {
            // Приховати рядок, якщо він не відповідає
умовам пошуку
            row.Visible = false;
        }
    }
}

```

```

    }
}

private void btnEdit_Click(object sender, EventArgs e)
{
    // Отримати вибраний рядок у DataGridView
    if (dataGridViewSuppliers.SelectedRows.Count > 0)
    {
        // Отримати дані товару з вибраного рядка
        string deliveryPostavshchuk =
dataGridViewSuppliers.SelectedRows[0].Cells["КодПоставщика"].Value
.ToString();

        string deliveryTovar =
dataGridViewSuppliers.SelectedRows[0].Cells["КодТовара"].Value.ToS
tring();

        string deliveryAmountTovar =
dataGridViewSuppliers.SelectedRows[0].Cells["КількістьТовара"].Val
ue.ToString();

        decimal deliveryPricePostavku =
Convert.ToDecimal(dataGridViewSuppliers.SelectedRows[0].Cells["Цін
аПоставки"].Value);

        string deliveryPruiniav =
dataGridViewSuppliers.SelectedRows[0].Cells["Прийняв"].Value.ToStr
ing();

        // Передати дані товару до форми редагування
        FormEditDelivery editForm = new
FormEditDelivery(deliveryPostavshchuk, deliveryTovar,
deliveryAmountTovar, deliveryPricePostavku, deliveryPruiniav);
        if (editForm.ShowDialog() == DialogResult.OK)
        {
            // Оновити дані товару в DataGridView

```

```
dataGridViewSuppliers.SelectedRows[0].Cells["КодПоставщика"].Value  
= editForm.DeliveryPostavshchuk;
```

```
dataGridViewSuppliers.SelectedRows[0].Cells["КодТовара"].Value =  
editForm.DeliveryTovar;
```

```
dataGridViewSuppliers.SelectedRows[0].Cells["КількістьТовара"].Val  
ue = editForm.DeliveryAmountTovar;
```

```
dataGridViewSuppliers.SelectedRows[0].Cells["ЦінаПоставки"].Value  
= editForm.DeliveryPricePostavku;
```

```
dataGridViewSuppliers.SelectedRows[0].Cells["Прийняв"].Value =  
editForm.DeliveryPruiniav;
```

```
    }
```

```
    }
```

```
    else
```

```
    {
```

```
        MessageBox.Show("Будь ласка, виберіть поставку для  
редагування.");
```

```
    }
```

```
    }
```

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

```
{
```

```
    new FormAddDelivery().Show();
```

```
}
```

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

```
{
```

```
    if (dataGridViewSuppliers.SelectedRows.Count > 0)
```

```

        {
            DialogResult result = MessageBox.Show("Ви дійсно
бажаєте видалити цю поставку(накладку)?", "Підтвердження
видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)
            {
                int selectedRowIndex =
dataGridViewSuppliers.SelectedRows[0].Index;
                int deliveryId =
Convert.ToInt32(dataGridViewSuppliers.Rows[selectedRowIndex].Cells
["КодПоставки"].Value);

                using (OleDbConnection connection = new
OleDbConnection(ConnectionString))
                {
                    try
                    {
                        connection.Open();

                        string query = "DELETE FROM Поставка
WHERE КодПоставки = @DeliveryId";
                        OleDbCommand command = new
OleDbCommand(query, connection);

                        command.Parameters.AddWithValue("@DeliveryId", deliveryId);

                        int rowsAffected =
command.ExecuteNonQuery();

                        if (rowsAffected > 0)
                        {
                            MessageBox.Show("Поставка(накладна) успішно видалений.");
                        }
                    }
                }
            }
        }
    }
}

```

```

        LoadProducts(); // Оновити дані
клієнтів в dataGridViewClients
    }
    else
    {
        MessageBox.Show("Помилка видалення
товара.");
    }
}
catch (Exception ex)
{
    MessageBox.Show("Помилка підключення
до бази даних: " + ex.Message);
}
}
}
else
{
    MessageBox.Show("Виберіть поставку (накладку), яку
бажаєте видалити.");
}
}

private void buttonPrint_Click(object sender, EventArgs e)
{
    if (dataGridViewSuppliers.SelectedRows.Count > 0)
    {
        // Отримати дані товару з вибраного рядка
        string deliveryPostavshchuk =
dataGridViewSuppliers.SelectedRows[0].Cells["КодПоставщика"].Value
.ToString();

        string deliveryTovar =
dataGridViewSuppliers.SelectedRows[0].Cells["КодТовара"].Value.ToS
tring();
    }
}

```

```

        string deliveryAmountTovar =
dataGridViewSuppliers.SelectedRows[0].Cells["КількістьТовара"].Val
ue.ToString();

        decimal deliveryPricePostavku =
Convert.ToDecimal(dataGridViewSuppliers.SelectedRows[0].Cells["Цін
аПоставки"].Value);

        string deliveryPruiniav =
dataGridViewSuppliers.SelectedRows[0].Cells["Прийняв"].Value.ToStr
ing();

        // Створити об'єкт класу PrintDocument
PrintDocument printDocument = new PrintDocument();

        // Налаштувати параметри друку
printDocument.DefaultPageSettings.PaperSize = new
PaperSize("Custom", 300, 400); // Задати розмір сторінки
printDocument.DefaultPageSettings.Margins = new
Margins(50, 50, 50, 50); // Задати поля
printDocument.DefaultPageSettings.Landscape =
true; // Режим альбомної орієнтації

        // Визначити обробник події PrintPage для
малювання даних друку
printDocument.PrintPage += (s, args) =>
{
    Graphics graphics = args.Graphics;
    int startX = 50;
    int startY = 50;
    int offset = 30;
    int lineHeight = 20;

    // Малювати заголовок накладної
    SizeF textSize =
graphics.MeasureString("Накладна", new Font("Arial", 16,
FontStyle.Bold));

```

```

        float textX = startX +
(printDocument.DefaultPageSettings.PrintableArea.Width -
textSize.Width) / 2;
        graphics.DrawString("Накладна", new
Font("Arial", 16, FontStyle.Bold), Brushes.Black, textX, startY);
        startY += offset;

        // Малювати дані друку
        graphics.DrawString("Поставщик: " +
deliveryPostavshchuk, new Font("Arial", 12), Brushes.Black,
startX, startY);

        startY += offset;
        graphics.DrawString("Назва Товара: " +
deliveryTovar, new Font("Arial", 12), Brushes.Black, startX,
startY);

        startY += offset;
        graphics.DrawString("Кількість: " +
deliveryAmountTovar, new Font("Arial", 12), Brushes.Black, startX,
startY);

        startY += offset;
        graphics.DrawString("Ціна товару: " +
deliveryPricePostavku.ToString("C"), new Font("Arial", 12),
Brushes.Black, textX, startY + offset);
        graphics.DrawString("Прийняв накладну: " +
deliveryPruiniav, new Font("Arial", 12), Brushes.Black, startX,
startY);

        startY += offset;

        args.HasMorePages = false;
};

// Викликати метод Print для початку друку

PrintDialog printDialog = new PrintDialog();

```

```

        printDocument.PrinterSettings =
printDialog.PrinterSettings;

        if (printDialog.ShowDialog() == DialogResult.OK)
        {
            PrintPreviewDialog printPreviewDialog = new
PrintPreviewDialog();
            printPreviewDialog.Document = printDocument;
            printPreviewDialog.ShowDialog();
            printDocument.Print();
        }
    }
else
    {
        MessageBox.Show("Будь ласка, виберіть накладну для
друку.");
    }
}

private void dataGridViewSuppliers_CellContentClick(object
sender, DataGridViewCellEventArgs e)
{
}
}
}

```