

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДІСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ
ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**«Інформаційна система управління продажами
комерційного підприємства»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Шелехов І.В.

Студентки групи ІН.мдн-92С

Лабунська А.В.

Суми 2021

Сумський державний університет
(назва вузу)

Факультет ІЗДВН Кафедра Комп'ютерних наук
Спеціальність 122 «Комп'ютерні науки»

Затверджую:
зав. кафедри _____

“ _____ ” _____ 20__ р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ**

Лабунській Анастасії Віталіївни
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна система управління продажами комерційного підприємства

затверджую наказом по інституту від “ _____ ” _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми та постановка задачі. 2) Проектування інформаційної системи. 3) Реалізація інтелектуальної системи

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

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1	<i>Аналіз проблеми та постановка задачі</i>		
2	<i>Проектування інформаційної системи</i>		
3	<i>Реалізація інтелектуальної системи</i>		
4	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Студент – дипломник _____

(підпис)

Керівник проекту _____

(підпис)

РЕФЕРАТ

Записка: 53 стор., 21 рис., 1 табл., 1 додаток, 47 джерел.

Предмет дослідження – автоматизація обліку реалізації продукції приватного підприємства.

Об'єкт дослідження – облік реалізації продукції приватного підприємства.

Мета роботи — розробка та програмна реалізація інформаційної системи управління продажами комерційного підприємства

Методи дослідження — технологія проектування інформаційної систем управління.

Результати — розроблено автоматизовану систему інформаційної (АІС) підтримки клієнтів. Для досягнення поставленої мети було виконано такі задачі: проведено аналіз предметної області, розроблено концептуальну модель даних АІС, розроблено логічну модель даних АІС, розроблено інфологічну модель даних АІС, визначено архітектуру АІС, виконано програмну реалізацію та тестування АІС.

АВТОМАТИЗАЦІЯ ОБЛІКУ, АВТОМАТИЗОВАНА СИСТЕМА
ІНФОРМАЦІЙНОЇ ПІДТРИМКИ КЛІЄНТІВ, МОДЕЛЬ ДАНИХ

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Організаційно-економічна сутність задачі	6
1.2 Опис вхідної та вихідної інформації	12
1.3 Постановка задачі.....	17
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	18
2.1 Розробка концептуальної моделі даних	18
2.2 Розробка логічної моделі даних.....	19
2.3 Розробка інфологічної моделі даних.....	24
3 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ	25
3.1 Формування вимог до програми.....	25
3.2 Проектування архітектури програми	26
3.3 Розробка алгоритмів та кодування	26
ВИСНОВОК.....	35
СПИСОК ЛІТЕРАТУРИ.....	36
ДОДАТОК.....	40

ВСТУП

Найбільш ранні системи керування базами даних виникли в 1960-х роках у відповідь на комп'ютеризоване управління комерційними даними. Такі системи були порівняно простими порівняно з сучасними додатками керування базами даних. Проте і ранні та новітні системи керування баз даних (СКДБ) мають спільні важливі елементи. Центральний аспект СКДБ - це не програма, яка виконує певні обчислення, а дані. Сьогодні деякі найвідоміші корпорації цінні не завдяки своїм фізичним активам, а скоріше завдяки власній інформації. Уявіть собі банк без його даних про рахунки та клієнтів або систему соціальної мережі, яка втрачає зв'язки між своїми користувачами. Вартість таких компаній за таких обставин була би майже повністю втрачена. Таким чином, сучасні СКДБ використовуються для управління колекціями даних, які є дуже цінними, відносно великими та доступними для кількох користувачів та додатків, часто одночасно.

Перші програми керування базами даних мали лише прості, точно відформатовані, структуровані дані. Сьогодні такі програми можуть включати дані зі складними взаємозв'язками та більш змінною структурою. Сучасні СКДБ використовують загальні характеристики в структурі даних для підвищення ефективності, але також дозволяють зберігати слабко структуровані дані та дані, формати яких постійно змінюються. Як результат, система керування баз даних - це велика, складна програмна система, завдання якої полягає в управлінні великим, складним набором даних.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Організаційно-економічна сутність задачі

Головна мета комерційної діяльності – одержання прибутку через задоволення купівельного попиту при високій культурі торговельного обслуговування. Ця мета рівною мірою важлива як для організацій і підприємств, так і для окремих осіб, що здійснюють операції купівлі-продажу на ринку товарів і послуг .

Зміст основ комерційної діяльності включає такі напрямки:

- закупівля матеріально-технічних ресурсів промисловими підприємствами й товарів оптово-посередницькими й ін. торговельними підприємствами;
- планування асортиментів і збуту продукції на промисловими підприємствами;
- організація збуту продукції підприємствами – виготовлювачами;
- вибір найкращого партнера в комерційній діяльності;
- організація оптового продажу товарів і комерційне посередництво;
- роздрібна торгівля, як форма комерційно-посередницької діяльності.

Комерційні відносини можуть розвиватися в умовах економічної волі суб'єктів ділових відносин, що припускає володіння капіталом й уміння управляти фінансами. Так само, це передбачає орієнтацію на витяг максимально можливого для складних умов прибутку й найбільш вигідні способи її капіталізації, уміння управляти комерційним ризиком, формування таких організаційних структур комерції, які здатні пристосовуватися до мінливих умов, сприйнятливість до змін у потребах ринку, повна рівноправність партнерів. У той же час не можна вважати економічною волею в комерційній діяльності повну незалежність від інтересів і дій суб'єктів ринку, оскільки в ряді випадків для досягнення яких-небудь стратегічних цілей необхідний компроміс із діловими партнерами. Крім того, воля комерційних взаємин може обмежуватися умовами зовнішнього середовища, комерційною таємницею й іншими об'єктивними факторами.

Для досягнення поставлених задач необхідно керуватися наступними основними принципами:

- нерозривний зв'язок комерції із принципами маркетингу;
- гнучкість комерції, її спрямованість на облік постійно мінливих вимог ринку;
- уміння передбачати комерційні ризики;
- виділення пріоритетів;
- прояв особистої ініціативи;
- висока відповідальність за виконання прийнятих зобов'язань по торговельних угодах;
- націленість на досягнення кінцевого результату - прибутку.

Інформаційна модель автоматизації процесів обробки заявок представлена на рисунку 1.1 і являє собою схему, яка відображає перетворення інформаційних реквізитів від джерел інформації до її одержувачів або, іншими словами, процес обробки інформації в інформаційній системі.

Інформаційна модель автоматизації процесів реєстрації та обробки заявок є схемою, яка відображатиме перетворення інформаційних реквізитів від джерел інформації до її одержувачів або, іншими словами, процес обробки інформації в інформаційній системі.

У побудованій моделі можна виділити кілька логічних рівнів:

1) джерела інформації, які складаються з:

- інформаційної системи, в якій зберігаються заповнені довідники;
- фахівців відділу продажів, які вводять вихідні дані в ІС;

2) первинна інформація:

- документи «Заявки від клієнтів»;
- форми введення «Вхідні заявки клієнта»;

3) таблиці з первинними даними:

- таблиці «Клієнти»;
- таблиця «Користувачі»;
- таблиця «Товари»;

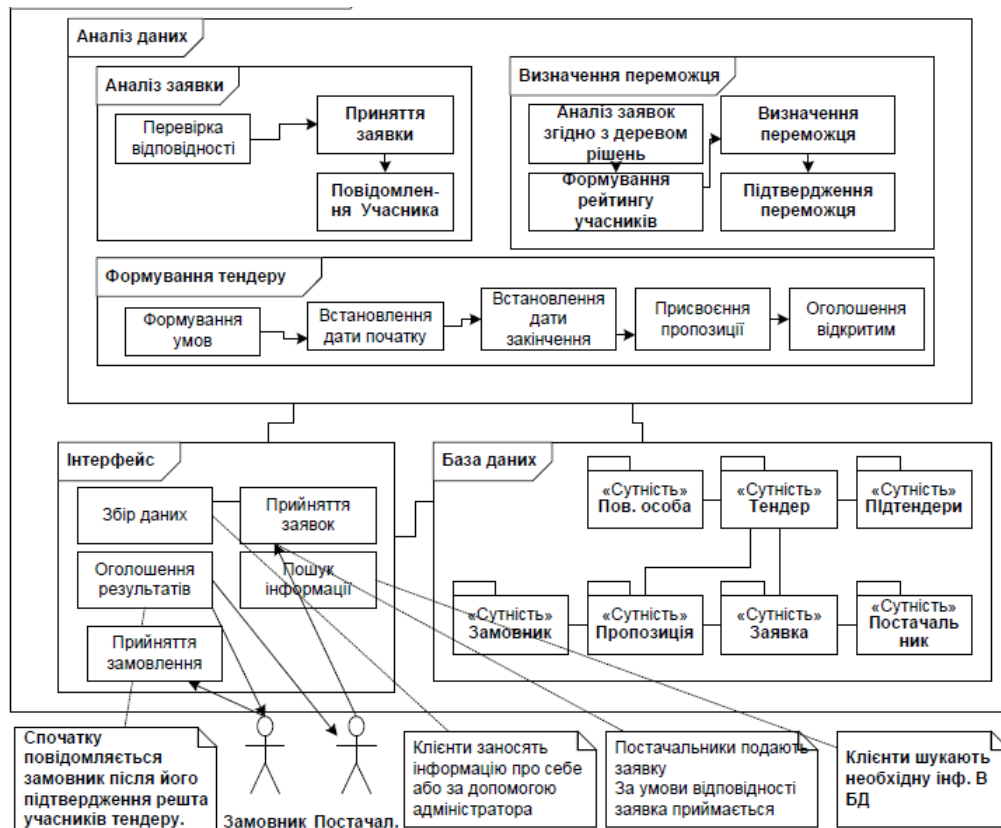


Рисунок 1.1 – Інформаційна модель системи реєстрації та обробки заявок на канцтовари

4) таблиці з проміжною і результативною інформацією:

- таблиця «Заявки клієнта»;

5) результатні документи і файли:

- документи «Товарна накладна»;
- документи «Рахунок на оплату»;

б) одержувачі інформації:

- підрозділи (Відділ продажів, Бухгалтерія, Керівництво компанії)[12].

Область інформаційної моделі відображає процес подачі заявки:

- таблиці «Заявки від клієнтів» заповнюється даними, що вводяться відділом продажів;

- співробітник відділ продажів, в разі якщо йому призначається заявка, в реальному часі отримує документ і може по ньому виконувати роботи;

Область 2 відображає процес ведення довідників ІС.

Для реалізації задачі обрано засоби СУБД.

Концептуальну схему тривіальної бази даних можна визначити безпосередньо шляхом аналізу системних вимог. Але в більших системах необхідна проміжна стадія - концептуальне проектування. Тому при моделюванні бази даних розроблювального проекту необхідно представити логічну модель бази даних на більше високому рівні абстракції, ніж у концептуальній схемі, тобто побудувати концептуальну модель бази даних.

Вона створюється за допомогою методів семантичного моделювання. Семантичне моделювання займається створенням моделей. При проектуванні бази даних ціль складається в створенні певного подання частини реального часу, що має сенс для його користувачів. Семантика вивчає зв'язки між словами й елементами реального часу, до яких ці слова ставляться. Семантика баз даних займається зв'язками між множиною даних й елементами реального часу, які воно представляє.

При моделюванні розроблювальної бази даних за основу модель сутність-зв'язок (ER-модель), що була вперше запропонована Ченом і з тих пір піддавалася значним уточненням. Ця модель має деякі недоліки, але не позбавлена достоїнств, які полягають у її відносній простоті а також в тому, що її зручно застосовувати до бізнес - сценаріїв. Саме різні варіації цієї моделі найчастіше використовуються при моделюванні сучасних баз даних.

Отже важливим етапом даної курсової роботи є розробка діаграми відносин між об'єктами, або, як її ще називають, діаграми "сутність-зв'язок" (Entity Relationship Diagram- ERD). Так, як необхідно зберігати інформацію про співробітників готелю, то логічніше всього було б представити її в відповідній таблиці. Це ж відноситься і до інформації про клієнтів готелю. Процес визначення структур даних, необхідних для зберігання інформації проекту, називається моделюванням даних.

Моделювання даних - це процес наочного зображення схеми бази даних, потоків даних, а також способу зв'язку окремих таблиць бази даних. Моделювання даних здійснюється із застосуванням різних засобів, включаючи діаграму "сутність-зв'язок" (Entity Relationship Diagram- ERD).

Діаграма "сутність-зв'язок" є однією із ключових концепцій, що застосовуються в процесі розробки баз даних. За допомогою цієї діаграми структуру даних було ретельно спроектовано, перш ніж модель була реалізована у вигляді бази даних. Діаграма відносин між об'єктами містить у собі логічні об'єкти, які були використані для подання в базі даних об'єктів реального часу, і взаємини між цими логічними об'єктами. У розроблювальному проекті діаграма "сутність-зв'язок" визначає взаємини між співробітниками готелю і його клієнтів.

Найголовніша частина цього процесу - чітке вираження ідеї, покладеної в основу проекту бази даних.

Структура бази даних розглядається з двох сторін: логічної й фізичної. Логічна структура бази даних визначає структуру таблиць, взаємини між ними, список користувачів, збережені процедури, правила й інші об'єкти бази даних. Фізична структура бази даних містить у собі опис файлів бази даних і журналу транзакцій, їхній первісний розмір, крок приросту, максимальний розмір, параметри конфігурації й т.д. На фізичному рівні визначаються способи зберігання даних з урахуванням подробиць і доступу до них. Сервер СУБД реалізує саме цей рівень.

Бази даних класифікують за моделями (або структурами) даних.

Модель даних — сукупність структур даних й операцій по їхній обробці. За допомогою моделі даних можна наочно представити структуру об'єктів і встановлені між ними зв'язку. Для термінології моделей даних характерні поняття «елемент даних» й «правила зв'язування». Елемент даних описує будь-який набір даних, а правила зв'язування визначають алгоритми взаємозв'язку елементів даних. До теперішнього часу розроблена множина різних моделей даних, але на практиці використовується три основних. Виділяють ієрархічну, мережну й реляційну моделі даних.

Розвиток реляційних баз даних почався наприкінці 60-х років, коли з'явилися перші роботи, у яких обговорювалися можливості використання при проектуванні баз даних звичних способів представлення інформації.

Базова реляційна модель досить проста, хоча розуміння цього приходить тільки після її вивчення. Реляційна модель призначена для опису певних відносин, що склалися між двома сутностями. У деяких базах даних ці відносини можуть приймати досить складну форму, однак якщо проаналізувати базу даних, розклавши її на складові частини, то по способу визначення зв'язків між сутностями можна легко виявити логіку роботи такої бази даних. База даних будується на сутностях й їхніх властивостях, які приймають відповідно форму рядків і стовпців таблиці.

Отже, реляційна модель служить для визначення відносин між сутностями. Реляційна модель лише описує спосіб зберігання й перегляду даних. Тобто, реляційна модель служить для завдання структури даних і визначення взаємин між наборами даних.

Основна ідея реляційної моделі даних полягає в тому, щоб представити будь-який набір даних у вигляді двовимірної таблиці. Реляційна модель описує структуру й взаємини між декількома різними таблицями.

Реляційною вважається така база даних, у якій всі дані представлені для користувача у вигляді прямокутних таблиць значень даних, і всі операції над базою даних зводяться до маніпуляцій з таблицями.

Метою бази даних є обробка даних про об'єкти реального часу, з урахуванням зв'язків між об'єктами. У теорії баз даних дані часто називають атрибутами, а об'єкти - сутностями. Об'єкт, атрибут і зв'язок - фундаментальні поняття.

Об'єкт - елементи реального часу, які існують незалежно.

Атрибут - це деякий показник, що характеризує якийсь об'єкт і приймає для конкретного екземпляра об'єкта деяке числове, текстове або інше значення. набором об'єктів, що мають власні атрибути.

Зв'язок представляє взаємодію між сутностями. Відношення між сутностями - це віртуальний зв'язок, заснований на взаємовідношеннях об'єктів реального часу. Відносини приймають одну із трьох форм: "один до одного", "один до багатьох" й "багато до багатьох". На діаграмі вона зображується

ромбом, що з'єднує сутності, що беруть участь у зв'язку. Зв'язок характеризується ступенем, що показує, скільки сутностей бере участь у зв'язку.

Одним з найкращих способів опису реляційної моделі є її практичне застосування на прикладі розроблюваної бази даних. Необхідно визначити сутності і їхні атрибути, які повинні будуть присутнім у проєктованій базі даних, та встановити існуючі між ними зв'язки.

1.2 Опис вхідної та вихідної інформації

Під вхідною оперативною інформацією розуміється вся інформація, необхідна для вирішення завдання і розташована на різних носіях: первинних документах, машинних носіях, в пам'яті персонального комп'ютера.

Від раціональної організації вхідної інформації оптового підприємства, способів збору, реєстрації, передачі, зберігання і обробки інформації, її складу і своєчасного отримання залежать оперативність і ефективність управління торговими процесами.

Вхідною інформацією для розроблюваної в дипломному проєкті автоматизованої системи є:

- для відображення прибуткових операцій прибуткові накладні, заповнюються від руки (форми вхідних документів приходу наведені в додатку). Ці документи необхідні для обліку надходження товару на склад від постачальників, а також для обліку надлишків, що виникають на складі.

- для відображення видаткових документів: товарно-транспортна накладна і рахунок-фактура, які заповнюються від руки. Ці документи підтверджують факт відвантаження продукції зі складу.

Для оформлення операцій заготівлі та придбання матеріалів використовуються розрахункові документи постачальників: рахунки-фактури, товарно-транспортні накладні та інші документи[4].

В системі задіяні такі довідники:

- 1) Довідник «Клієнти» - містить інформацію про клієнтів компанії.
- 2) Довідник «Користувачі» - містить загальну інформацію про співробітників компанії, які є користувачами системи.

3) Довідник «Товари» - містить інформацію про товари, які необхідні клієнту.

Таблиця 1.1 – Актуалізація довідників системи

Назва довідника	відповідальний	Середній обсяг довідника, записів	Середня частота актуалізації	Середній обсяг актуалізації
Довідник «Клієнти»	Відділ продажів	1000	5 разів на день	10 записів
Довідник «Товари»	Відділ продажів	5000	1 раз на тиждень	10 записів
Довідник «Користувачі»	Відділ продажів	10	1 раз на рік	1-2 записи

В даному проєкті результуючою інформацією є 2 звіти, які беруть дані з таблиці заявок або довідників:

1. «Товарна накладна»

Формується на основі таблиць і полів:

- Номер заявки
- Дата створення заявки
- Клієнт (заявник)
- Призначений на заявку співробітник
- Статус заявки
- Найменування товару

Даний звіт служить для отримання статистики роботи відділу продажів, а так само для оперативного управління і прийняття рішень. У підсумках даного звіту за місяць можна зрозуміти, наскільки завантажений відділ продажів при реєстрації заявок, з якими проблемами стикаються клієнти.

2. Звіт «Рахунок на оплату»

Містить таблицю з наступними полями:

- Номер заявки
- Дата створення заявки
- Виконані роботи
- Вартість виконаних робіт

Даного роду звіт служить для виставлення рахунку на оплату клієнтові в разі виконання робіт за заявкою клієнта.

Виходячи з потреб дійових осіб, виділяються наступні варіанти використання (рисунок 1.2):

- перевірити інформацію про замовлення на відвантаження матеріалів;
- перевірити інформацію про оформлення вхідних документів;
- сформувати графік динаміки руху матеріалів;
- сформувати відомість залишків матеріалів на складі;
- сформувати відомість обліку матеріалів[9].

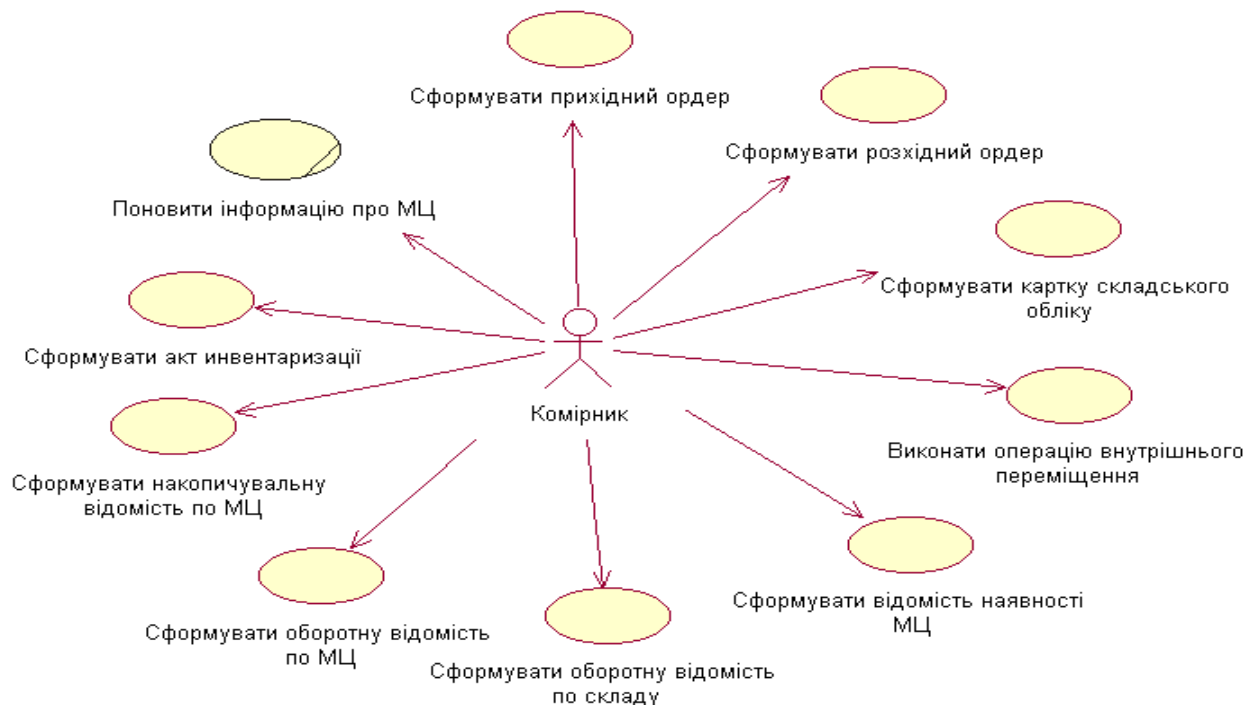


Рисунок 1.2 – Діаграма варіантів використання комплексу задач “Облік матеріалів”

Варіант використання «Перевірити інформацію про замовлення на відвантаження матеріалів» Дозволяє комірнику одержати інформацію про замовлення на відвантаження матеріалів, а також внести необхідні зміни.

Основний потік подій. Даний варіант використання починає виконуватися, коли комірник вводить запит на отримання інформації про замовлення та їх стан. Система пропонує можливість виведення інформації на поточну дату або за будь-

який період і виводить відомості про замовлення. Комірник може внести необхідні зміни; наприклад, якщо поступила інформація про те, що матеріал відвантажено замовнику, то замовлення вважається виконаним; комірник може видалити замовлення, якщо були порушені правила оформлення вхідних документів, і запобігання цим порушенням не передбачено в системі.

Альтернативні потоки. Якщо з'єднання із сервером неможливе, то видається відповідне повідомлення.

Передумови. Замовники оформляють вхідні документи на отримання матеріалу за допомогою підсистеми реєстрації замовлень.

Постумови. Якщо варіант використання завершиться успішно, комірником буде отримано (створено, оновлено, видалено) перелік замовлень, що є основою руху матеріалів на складі. Інакше стан системи не зміниться.

Варіант використання «Перевірити інформацію про оплату»

Короткий опис. Система обновляє відомості про оплату замовлень клієнтами.

Основний потік подій. Замовлення вважається сплаченим тільки після підтвердження інформації про оплату через банк або платіжну систему.

Альтернативні потоки. Якщо з'єднання із сервером банку неможливе, то видається відповідне повідомлення.

Передумови. Необхідно з'єднається з сервером банку або платіжної системи.

Постумови. Після перевірки сплачені замовлення наголошуються і можуть бути доставлені.

Варіант використання «Сформувати відомість залишків матеріалів»

Короткий опис. Система формує відомість залишків матеріалів відповідно до введених користувачем параметрів.

Основний потік подій. Комірник вводить запит на формування відомості обліку та параметри запиту: період часу, категорії матеріалів і система формує відомість.

Альтернативні потоки. Якщо відповідно до заданих параметрів відсутні дані про матеріали, то виводиться відповідне повідомлення.

Передумови. Відомість формується тільки при умові, якщо виконувалися відвантаження матеріалів за вказаний період, якщо ні, то користувач може змінити параметри запиту.

Постумови. Формування документа для обліку поточної діяльності. При необхідності можна роздрукувати документ.

Варіант використання «Сформувати відомість обліку матеріалів»

Короткий опис. Система формує відомість обліку матеріалів відповідно до введених користувачем параметрів.

Основний потік подій. Комірник вводить запит на формування відомості обліку і вводить параметри запиту: період часу, категорії матеріалів і система формує відомість.

Альтернативні потоки. Якщо відповідно до заданих параметрів відсутні дані про матеріали, то виводиться відповідне повідомлення

Передумови. Відомість формується тільки при умові, якщо виконувалися відвантаження матеріалів за вказаний період, якщо ні, то користувач може змінити параметри запиту.

Постумови. Формування документа для обліку поточної діяльності. При необхідності можна роздрукувати документ[3,4].

Проміжна інформація формується в процесі вирішення деяких розроблених завдань на ЕОМ і використовується для виконання відповідних розрахунків. До такої інформації слід віднести інформацію, що зберігається в таблицях:

- Залишки;
- Журнал;

Дані в таблиці Залишки оновлюються з кожним приходом товару, тобто до кількості фактичного залишку додається кількість знову надходження товару. Оновлення відбувається за допомогою запиту Оновлення залишків по приходу при проведенні прибуткової накладної. А при кожному витратах товару оновлення відбувається шляхом віднімання від фактичної кількості товару кількості реалізованого товару за допомогою запиту Оновлення залишків по витраті при проведенні Товарної накладної.

Також і при оприбуткуванні товару на склад додається кількість оприбуткованого товару, а при списанні віднімається кількість списаного товару.

Проміжна інформація таблиці Залишки формується, використовуючи інформацію по замовленнях клієнта.

Дані таблиці Залишки є вихідними при формуванні звіту Залишки на складі.

Дані в таблицю Журнал додаються при кожній парафії і при кожному витратах товару, при оприбуткуванні і списання. У журналі реєструються всі документи по приходу, витраті, оприбуткуванню та списання товару, в ньому вказується дата операції, номер документа і контрагент.

При проведенні документа, дані автоматично додаються в таблицю Журнал з допомогою запитів Проведення приходу, Проведення витрати, Проведення оприбуткування, Проведення списання.

Дані таблиці Журнал використовуються як проміжні при формуванні звіту Журнал по приходу і Журнал по витраті.

1.3 Постановка задачі

Метою роботи є розробка автоматизованої системи інформаційної (АІС) підтримки клієнтів. Для досягнення поставленої мети необхідно виконати такі задачі дослідження:

- 1) аналіз предметної області,
- 2) розробка концептуальної моделі даних АІС,
- 3) розробка логічної моделі даних АІС,
- 4) розробка інфологічної моделі даних АІС,
- 5) проектування архітектури АІС,
- 6) реалізація та тестування АІС.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Розробка концептуальної моделі даних

У відповідність до проведеного аналізу було з'ясовано, що процес прийняття заявки від клієнта в компанії продажу продукції здійснюється на підставі статуту, внутрішніх положень, посадових інструкцій співробітників відділів відповідальних за ведення заявок.

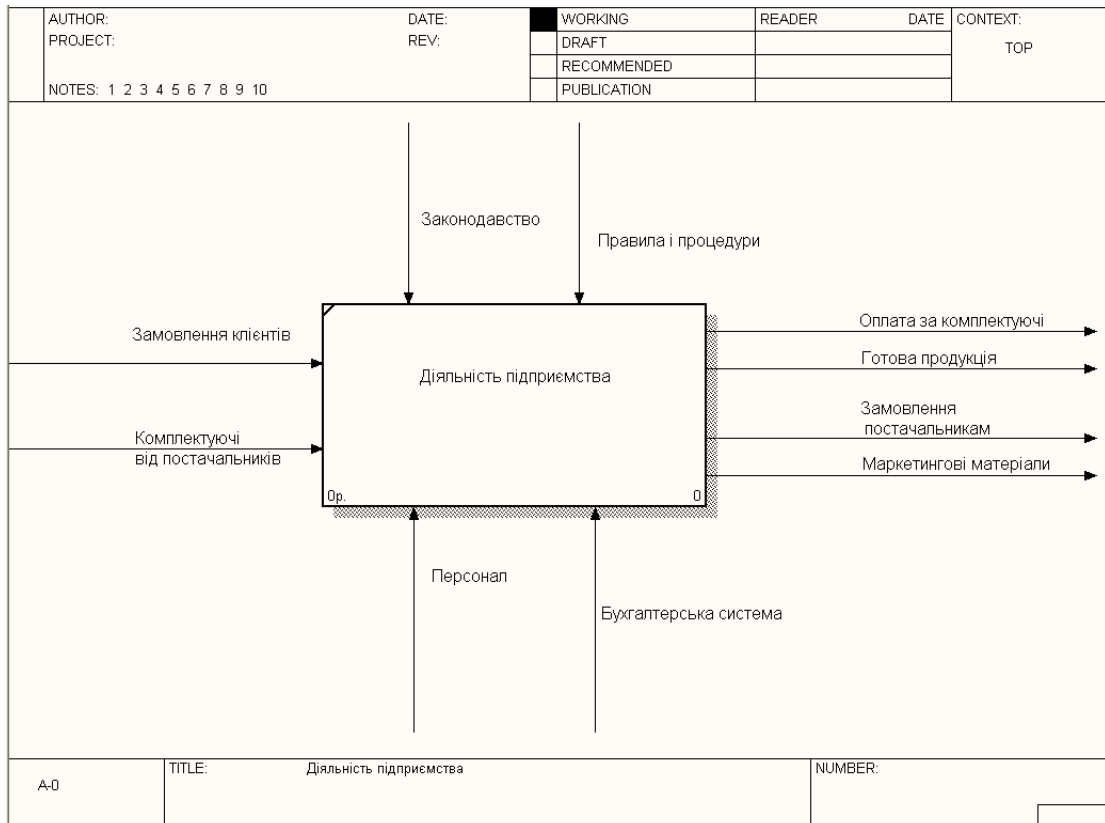


Рисунок 2.1 – Контекстна діаграма процесу прийняття заявки від клієнта на продаж продукції (AS-IS)

У процесі реєстрації заявки від клієнта на рішення проблем беруть участь зовнішні джерела, співробітники компанії (фахівці з продажу), адміністрація (директор, бухгалтер, начальник відділу продажів).

В результаті здійснення операцій з даними заявками може бути сформований журнал заявок клієнта, документ про виконані роботи по заявці клієнта, рахунок на оплату.

В процесі обробки вхідної документації обробляються і реєструються заявки, що надходять від зовнішніх джерел (клієнтів). При виконанні процесів дані документи реєструють і передають в якості документа-підстави для виконання робочих операцій.

У процесі формування заявок основна роль у виконанні даного бізнес-процесу покладається на співробітників відділу продажів.

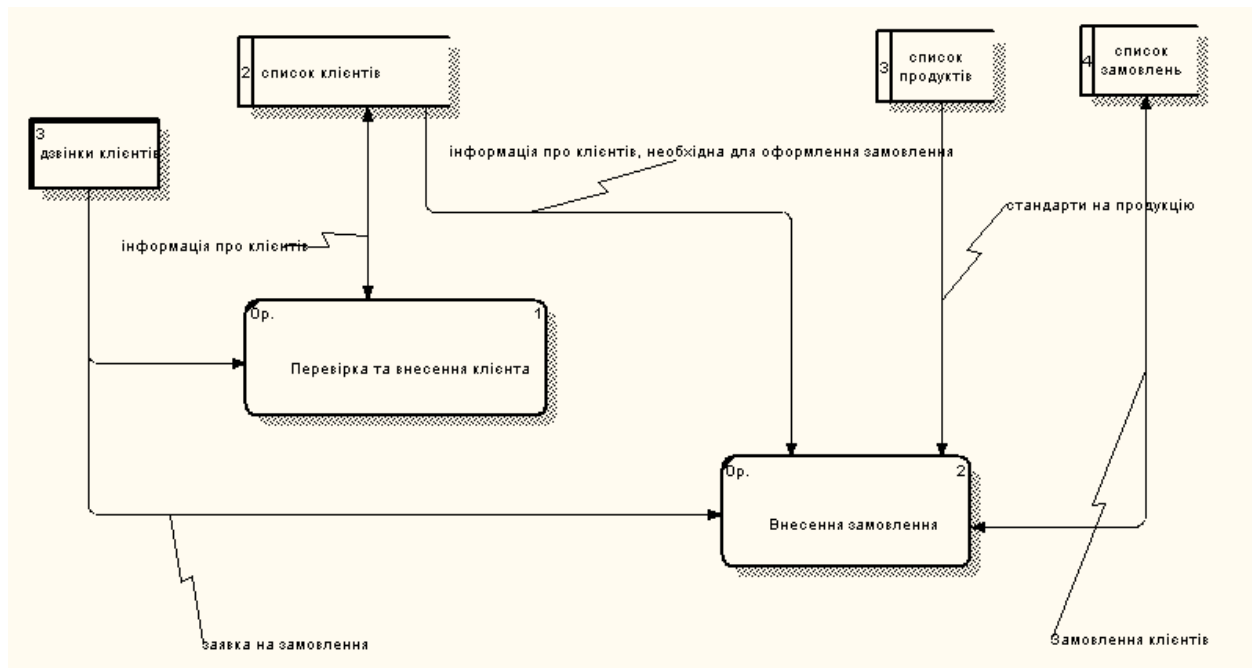


Рисунок 2.2 – Діаграма декомпозиції процесу прийняття заявки від клієнта

У процесі вирішення заявок від клієнтів розглядаються і приймаються плани робіт та звіти про виконану роботу. Основна роль у виконанні даних бізнес-процесів покладається на адміністрацію[3].

2.2 Розробка логічної моделі даних

Розроблюваний програмний продукт автоматизує функції обробки заявок компанією.

Виділяється два підмножини функцій системи реєстрації та обробки заявок:

- службові функції, до яких відносяться вхід в систему з обліковим записом користувача, додавання документів, допомога по роботі з програмою;

- основні функції обробки даних: ведення довідників системи, перегляд і друк документів.

Система повинна працювати в розрахованому на багато користувачів режимі. При вході в систему під певним обліковим записом, користувач отримує доступ до тих функцій системи, які йому доступні.

На рисунку 2.3 представлений сценарій діалогу інформаційної системи.

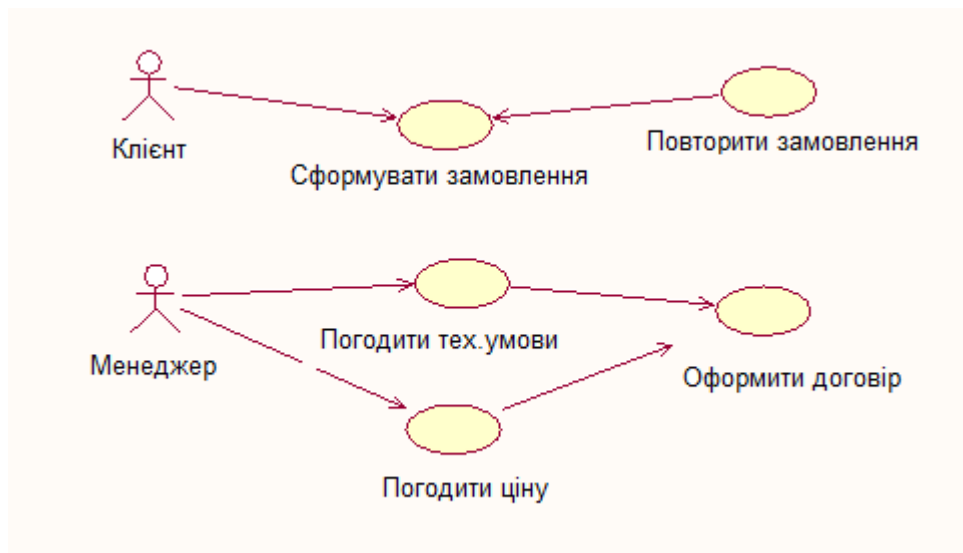


Рисунок 2.3 – Сценарій діалогу реєстрації та обробки заявок

Структура меню програми, що розробляється, складається з наступних модулів:

- службові (вхід в систему, зміна користувача, вихід з програми);
- заявки (заявки на товар);
- звіти (товарна накладна, рахунок на оплату);
- довідники (клієнти, користувачі, товар);
- довідка (про програму).

Першим кроком при створенні логічної моделі БД є побудова діаграми ERD (Entity Relationship Diagram, Діаграма сутність-зв'язок).

Об'єкти моделі, представлені на логічному рівні, називаються сутностями і атрибутами. Логічна модель даних є універсальною і ніяк не пов'язана з конкретною реалізацією СУБД.

Діаграма сутність-зв'язок є найвищим рівнем в моделі даних і визначає набір сутностей і атрибутів проектованої системи.

ERD-діаграма дозволяє розглянути систему цілком і з'ясувати вимоги, необхідні для її розробки, що стосуються зберігання інформації.

ERD-діаграми можна поділити на окремі шматки, відповідні окремим завдань, що вирішуються проектованої системою. Це дозволяє розглядати систему з точки зору функціональних можливостей, роблячи процес проектування керованим.

При створенні концептуальної моделі були визначені сутності й атрибути сутностей і складена логічна модель бази даних представлена на рисунку 2.4.

У логічній моделі бази даних встановлено три батьківські суті: «Матеріали», «Підрозділи» і «Склади», які пов'язані з ТМЦ зв'язками один до багатьох з ключових полях. У свою чергу зі складу ТМЦ видаються користувачам, що відображається як створення. Створення ідентифікує зв'язку один до багатьох. Це означає, що у всіх випадках один примірник першої сутності взаємодіє з декількома екземплярами іншої сутності. Взаємозв'язку відображаються лініями, що з'єднують дві сутності, Кожна сутність ділиться на 2 групи. У першій групі знаходяться атрибути, звані первинним ключем. Первинний ключ - це набір атрибутів, обраних для ідентифікації унікальних екземплярів сутності. Первинний ключ потрібен для того, щоб від нього створювати зв'язку між іншими таблицями.

При створенні суті необхідно виділити атрибути, які можуть стати первинним ключем (потенційні ключі), потім провести відбір атрибутів, слідуючи наступних рекомендацій:

1. Первинний ключ повинен бути підбраний таким чином, щоб за значеннями атрибутів, в нього включені, можна було точно ідентифікувати примірник суті;

2. Жоден із атрибутів первинного ключа не повинен мати нульове значення.

3. Значення атрибутів первинного ключа не повинні змінюватися. Якщо значення змінилося, значить, це вже інший екземпляр сутності.

На діаграмі поряд зі зв'язком відбивається її ім'я, що показує відношення між сутностями. При проведенні зв'язку між сутностями первинний ключ передається або мігрує в дочірню сутність.

На наступному етапі побудови логічної моделі визначаємо ключові атрибути і типи атрибутів. Типи атрибутів представлені в таблиці (номер накладної в даному випадку текстовий)[13].

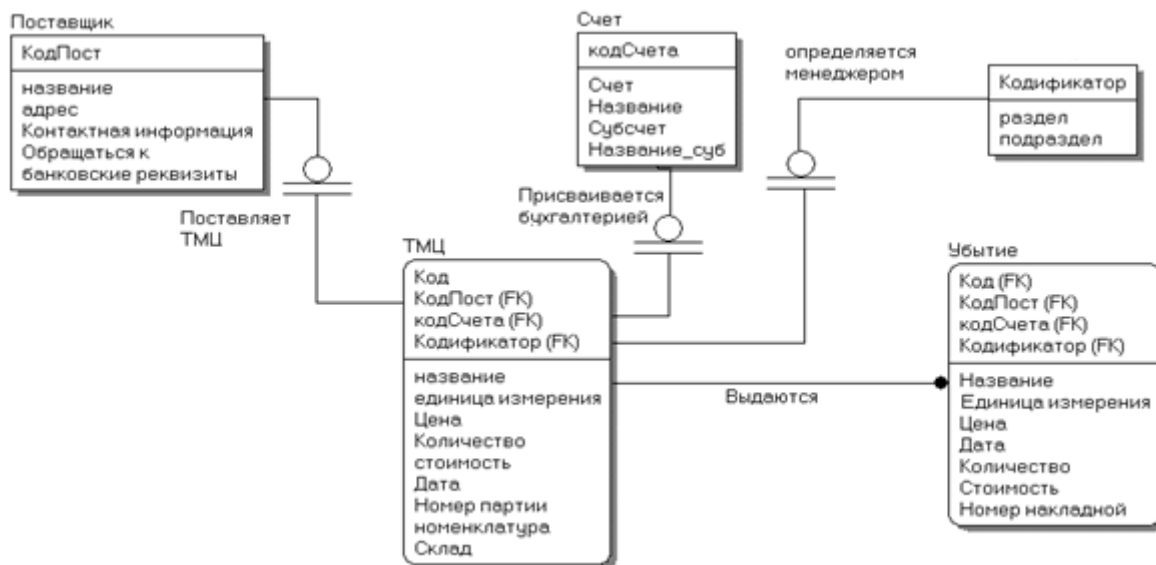


Рисунок 2.4 – Даталогічна модель

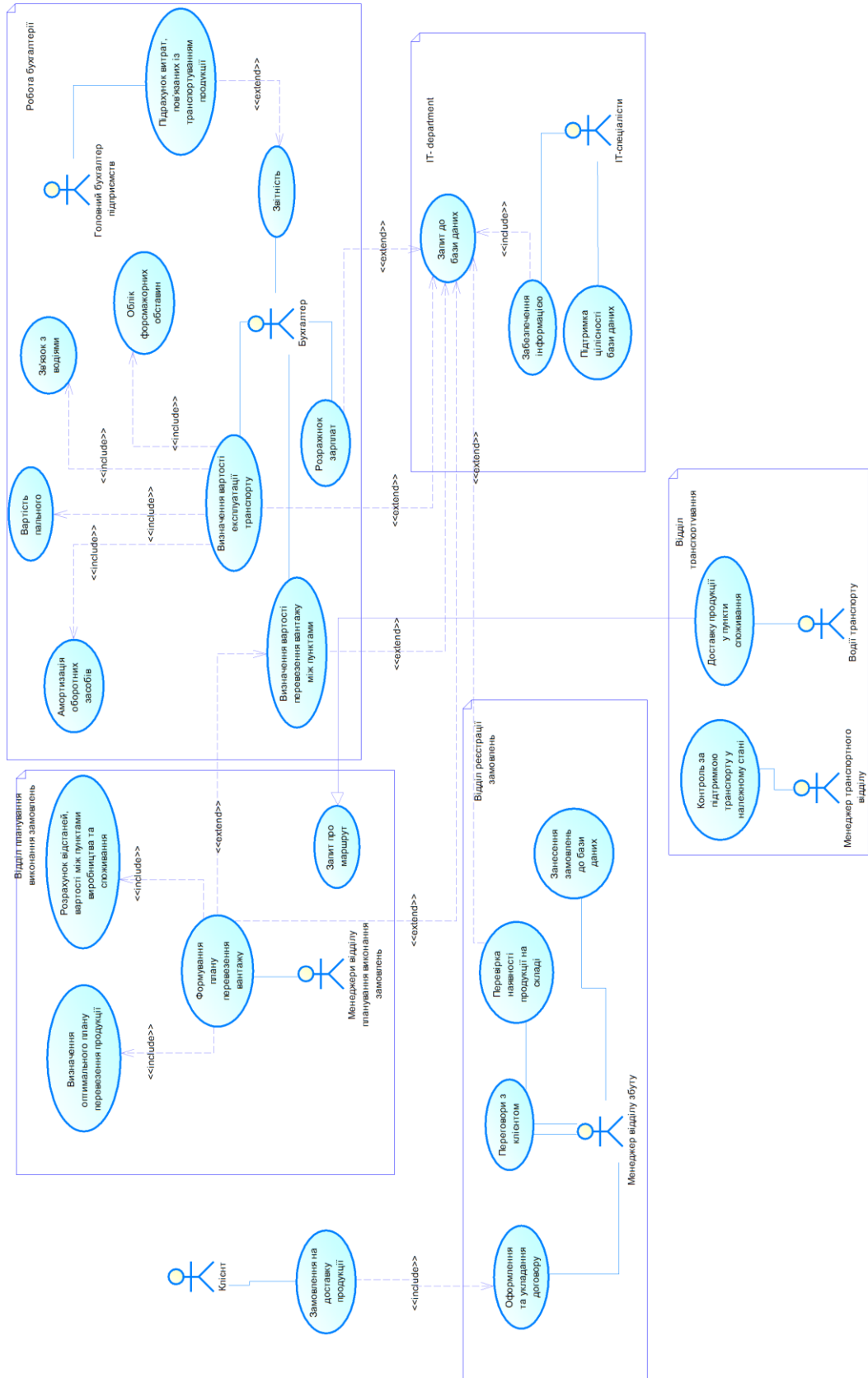


Рисунок 2.5 – Концептуальна модель

2.3 Розробка інфологічної моделі даних

Замовлення продукції. На рисунку 2.6 показана діаграма «сутність-зв'язків» та опис типів даних

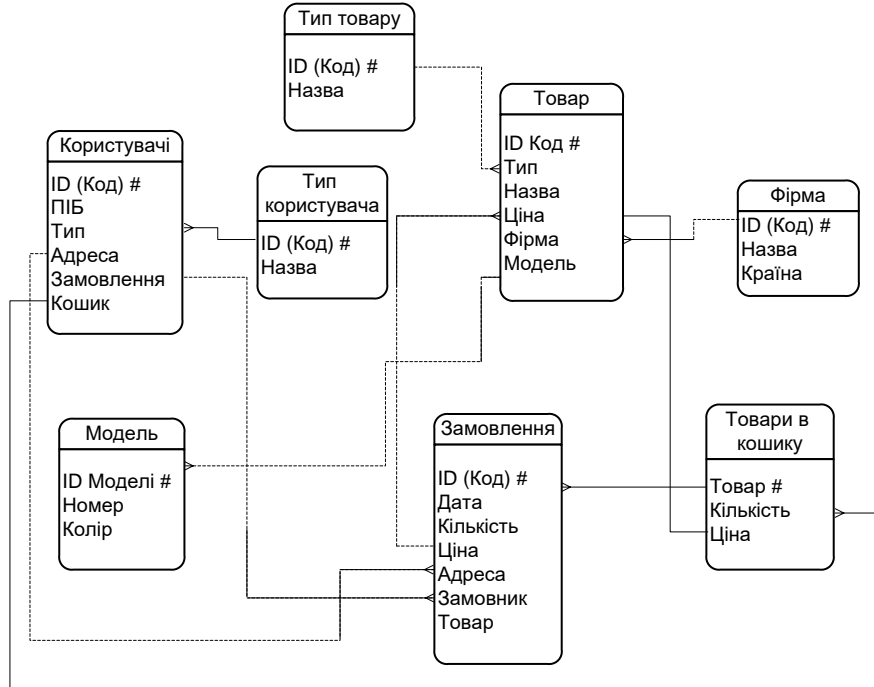


Рисунок 2.6 – Схема структурна «сутність-зв'язок» предметної області

Розроблена схема варіантів використання представлена на рисунку 2.7.

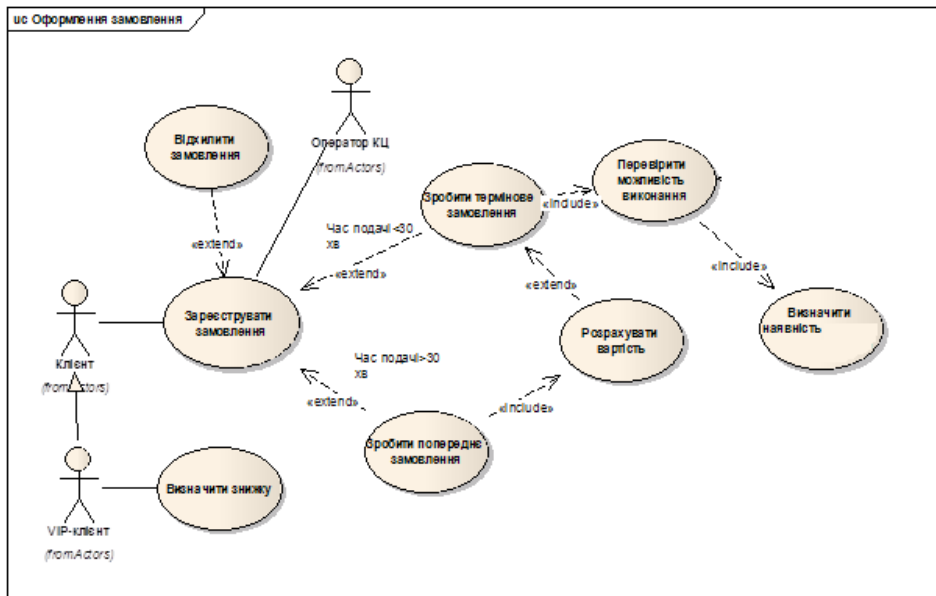


Рисунок 2.7 – Схема варіантів використання

3 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Формування вимог до програми

На сьогоднішній момент є бізнес процеси, що вимагають автоматизації і задіють відділ продажів, а саме процеси пов'язані з прийомом і обробкою заявок відділом продажів від клієнтів і подальша робота з ними.

Всі заявки ведуться фахівцями з продажу в програмі Excel. Для кожного клієнта в Excel виділено окремий лист, в якому знаходиться вся інформація про взаємини з клієнтом. Для кожного клієнта в Excel виділено окремий лист, в якому знаходиться вся інформація про переговори з клієнтом. Для того щоб знайти потрібну інформацію по клієнту який вже користувався послугами компанії в Excel зберігається посилання номер клієнта в програмі партнера [21].

Проблемами даної реалізації є:

- можливе дублювання або некоректний введення інформації в таблицю Excel, а також громіздкість представлення інформації і складність її пошуку;
- часові витрати під час завантаження форми Excel, а також відсутність моніторингу і стандартів заповнення даних про заявки клієнта.

Як наслідок інформація про заявки клієнтів може бути відсутня, бути вказана некоректно або не повністю. Документи заповнюються або вручну, або з шаблонів Word. При цьому дані для заповнення шаблону дублюються в Excel, що збільшує час створення документа і відповідно затримує клієнта.

Ще однією проблемою є те, що кожен з менеджерів компанії веде свій файл Excel зі своїми позначками по клієнтам, відповідно в разі хвороби, відпустки або звільнення іншому менеджеру потрібно ще більше часу, щоб розібратися з наявними даними. Причому найчастіше ними навіть і не користуються, вводячи все дані по клієнту заново.

Так само існує ймовірність втрати даних шляхом пошкодження файлу, а так само шляхом випадкового або навмисного видалення. До того ж немає можливості відстежити історію змін, тобто є ймовірність того що дані які вводив менеджер могли підмінити.

Все це призвело до того, що необхідно вдосконалювати існуючу систему реєстрації та обробки заявок від клієнтів в «Юкос».

На рисунку 2.1 показана контекстна діаграма ведення журналу заявок клієнтів.

У відповідність до проведеного аналізу було з'ясовано, що процес прийняття заявки від клієнта здійснюється на підставі статуту компанії, внутрішніх положень, посадових інструкцій співробітників відділів відповідальних за ведення заявок.

3.2 Проектування архітектури програми

За допомогою програмного продукту MySQL Workbench 5.2 CE побудована логічна модель даних на якій представленні головні таблиці, що зв'язані з контентом наповнення сайту, які знаходяться у базі даних (рисунок 2.5).

Система має наступні сутності, необхідні для вирішення задачі:

z95ca_virtuemart_vmuser_shoppengroups – містить списки користувачів та продавців;

z95ca_virtuemart_userinfos – містить інформацію про користувачів;

z95ca_virtuemart_vendors – містить інформацію про продавців системи;

z95ca_virtuemart_orders – містить інформацію про замовлення;

z95ca_virtuemart_categories – містить інформацію про категорії;

z95ca_virtuemart_products – містить інформацію про товари;

z95ca_virtuemart_products – Містить інформацію про відгуки до товарів;

z95ca_virtuemart_products_price – Містить інформацію про ціни товарів;

z95ca_virtuemart_auth_use_vendor – містить списки користувачів та продавців;

z95ca_virtuemart_product_reviews – Містить інформацію про відгуки до товарів.

3.3 Розробка алгоритмів та кодування

Задача модулю містить у собі рішення задач автоматизації та прискорення робочого процесу. За допомогою діаграм діяльності варіантів використання

можна детально ознайомитись з процесом роботи та діями що проходять всередині цих процесів [25, 2].

Діаграма діяльності варіанту використання «Автоматизація», містить в собі перелік подій, та необхідних дій для автоматизації в систему під своїм обліковим записом з використанням існуючих даних для введення їх у поля логіну та паролю (рисунок 3.1).

Діаграма діяльності варіанту використання «Додавання користувача», містить в собі перелік дій необхідних для вдалого виконання процесу додавання запису про користувача до бази даних (рисунок 3.2). При заповненні усіх необхідних полів для додавання запису, потрібно натиснути кнопку «Зберегти», після чого дані будуть внесені до бази даних.

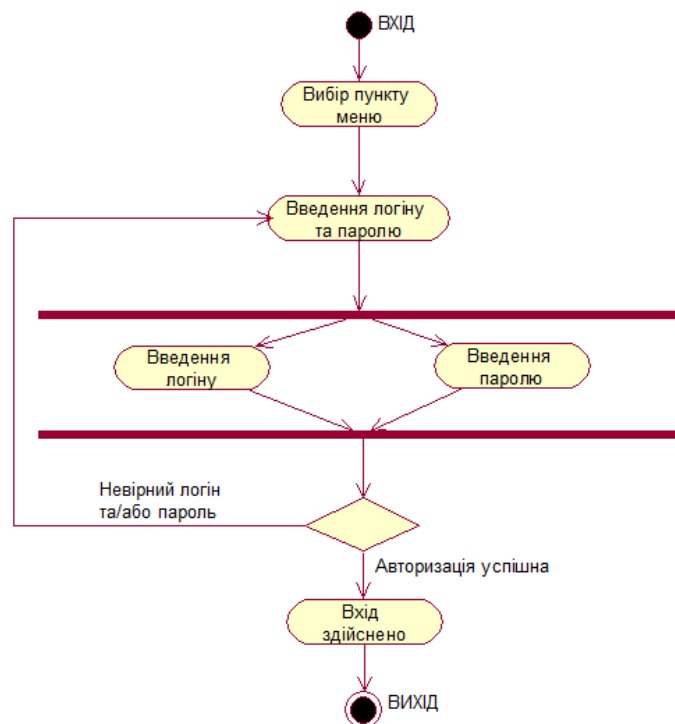


Рисунок 3.1 – Діаграма діяльності варіантів використання «Автоматизація»

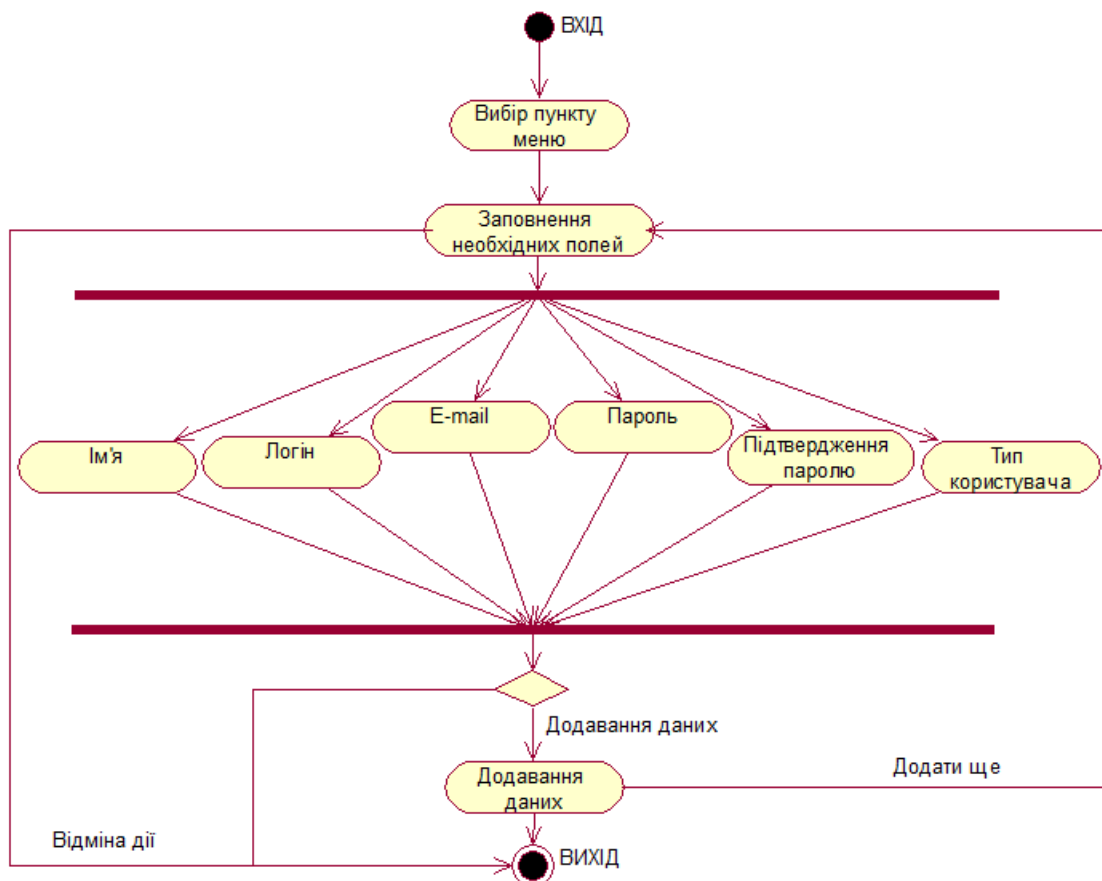


Рисунок 3.2 – Діаграма діяльності варіанту використання «Додавання користувача»

Діаграма діяльності варіанту використання «Редагування даних про користувача» (рисунок 3.3), містить перелік дій, необхідних для вдалого редагування вибраного запису у довіднику користувачів. При зміні усіх необхідних полів, треба натиснути кнопку «Зберегти», після чого зміни будуть внесені до бази даних.

Діаграма діяльності варіанту використання «Видалення запису про користувача» (рисунок 3.4) містить послідовність дій необхідних для видалення запису с бази даних. При спробі видалити запис, система запитає, чи справді Ви хочете видалити запис. При натисненні на кнопку «Так» запис буде видалений, інакше система перейде у режим очікування.

З цих діаграм видно, що модуль містить у собі рішення задач автоматизації та прискорює робочий процес. За допомогою діаграм діяльності варіантів

використання було проведено детальне ознайомлення з процесом роботи та діями що проходять всередині цих процесів.

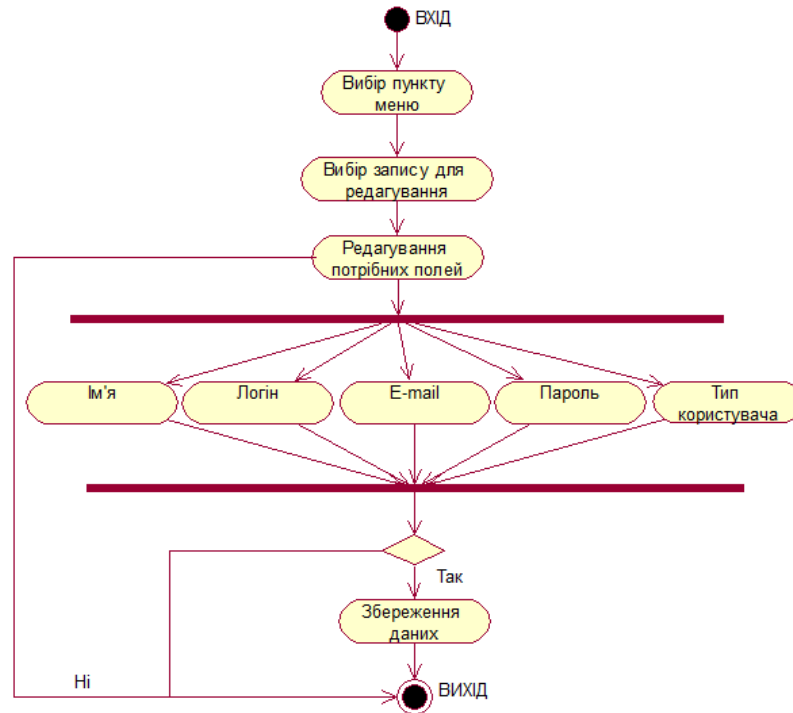


Рисунок 3.3 – Діаграма діяльності варіанту використання «Редагування даних про користувача»

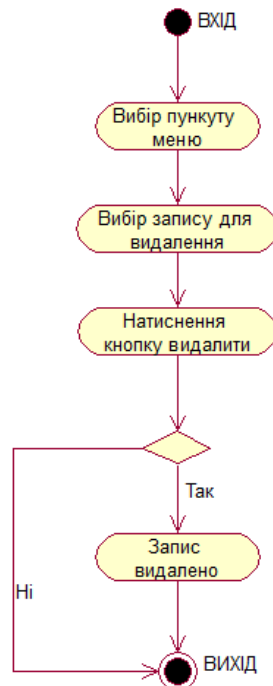


Рисунок 3.4 – Діаграма діяльності варіанту використання «Видалення запису про користувача»

Мета моделювання документів – описати атрибути документів, їх типи, значення, правила формування для проектування інтерфейсу системи, проектування бази даних системи; формування альбому вихідних форм системи.

Діаграма класів зображена на рисунок 3.7.

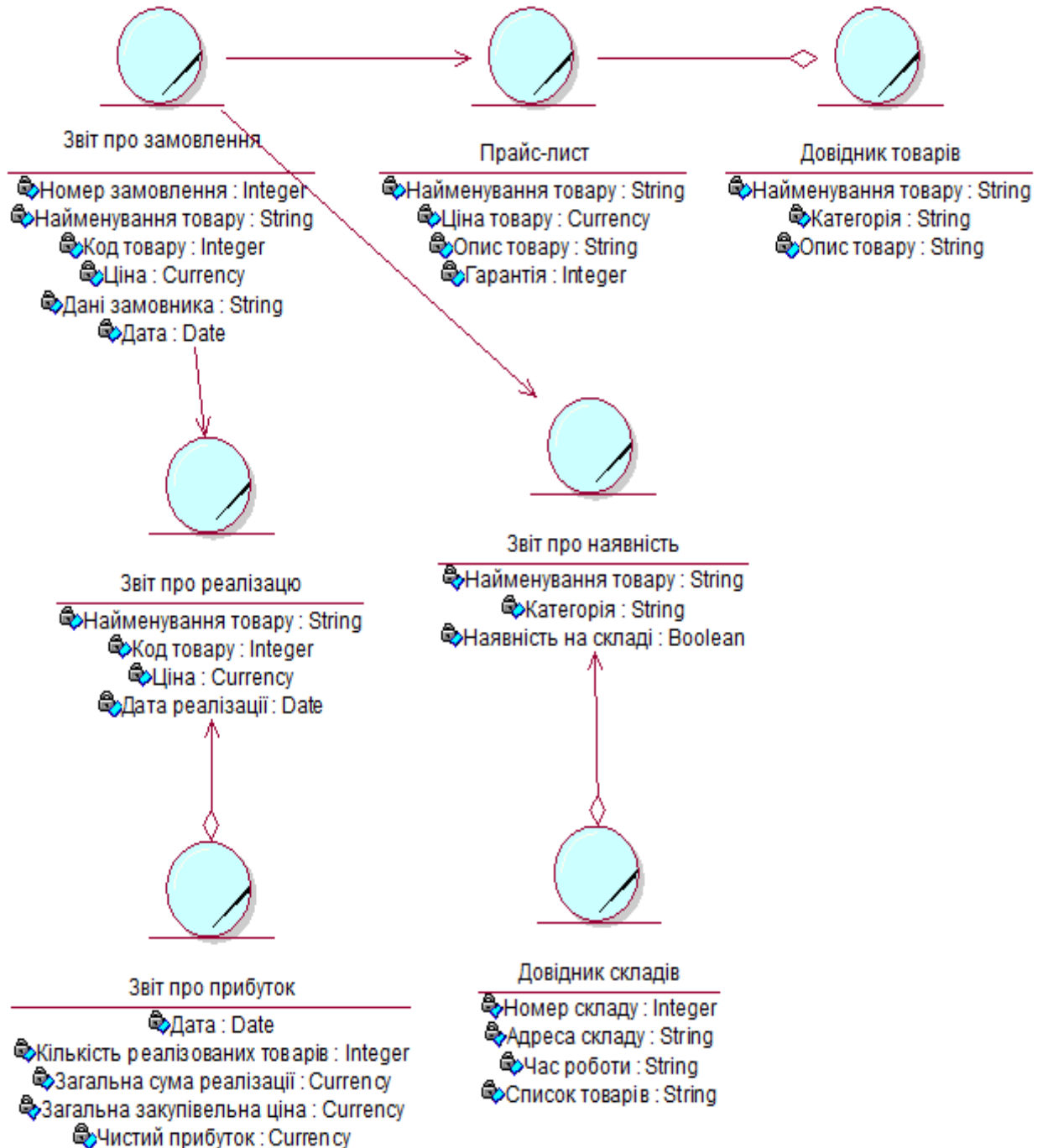


Рисунок 3.5 – Діаграма класів

Ярлик для запуску програми знаходиться на робочому столі ПК.

При запуску програми з'являється головне вікно програми з основними функціональними елементами:

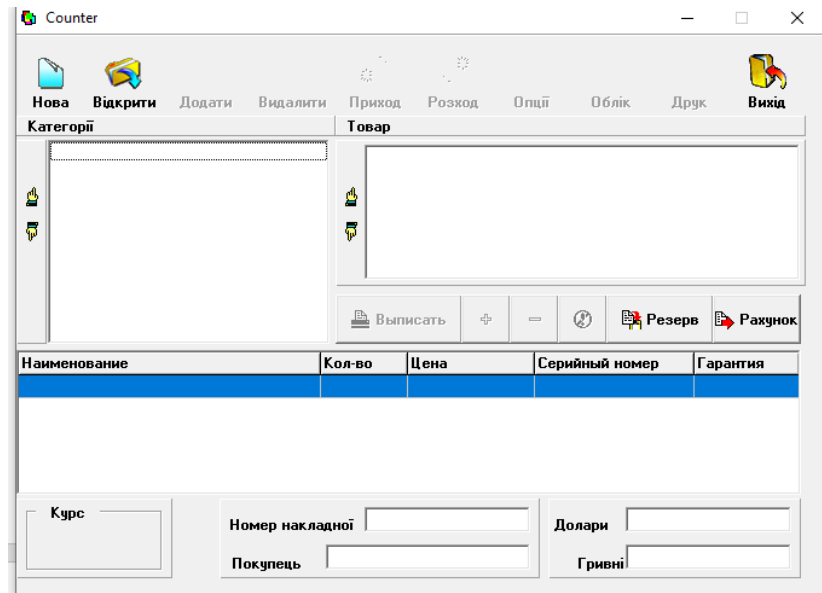


Рисунок 3.6 – Головне вікно програми

Для відкриття існуючої бази скористаємося кнопкою Відкрити головної форми, з'явиться вікно вибору бази даних:

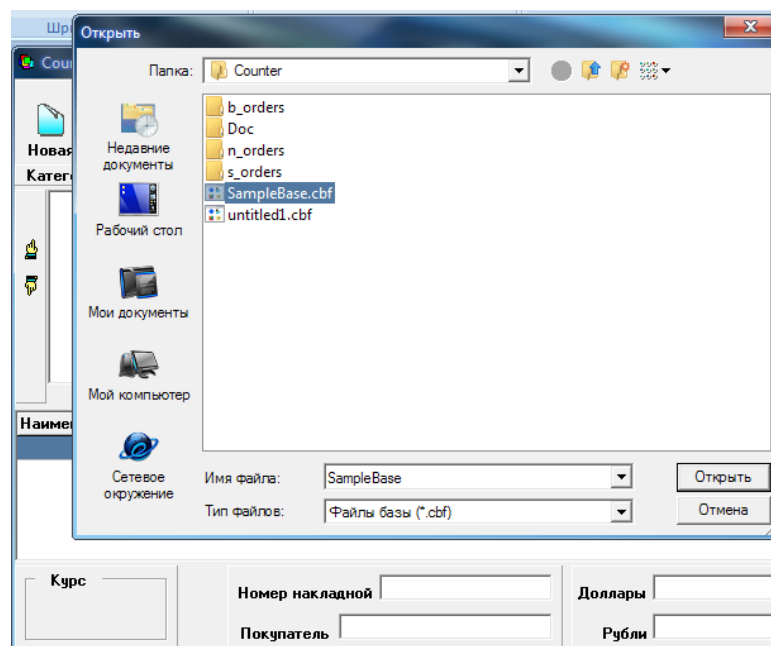


Рисунок 3.7 – Відкриття БД

При створенні або видаленні БД потрібно скористатися кнопками Створити і Видалити відповідно.

Для постановки товару на прихід використовується кнопка Прихід головного вікна, яка виводить форму для введення основних відомостей про куплений товар: ціни, номери накладної, дату закупівлі, контрагента, кількості.

Рисунок 3.8 – Форма «Прихід»

Для обліку продажів використовується кнопка головного вікна Витрата:

Рисунок 3.9 – Форма «Витрата»

Відкривається форма Продаж / Рахунок, де зазначається інформація про контрагента, типу продажу, кількості покупок і загальної вартості покупки, вказується накладна.

Для введення інформації про новий контрагента використовуємо кнопку Новий, де у вікні Клієнт вказуємо необхідну інформацію:

Рисунок 3.10 – Форма продажу товару

Рисунок 3.11 – Форма обліку клієнтів

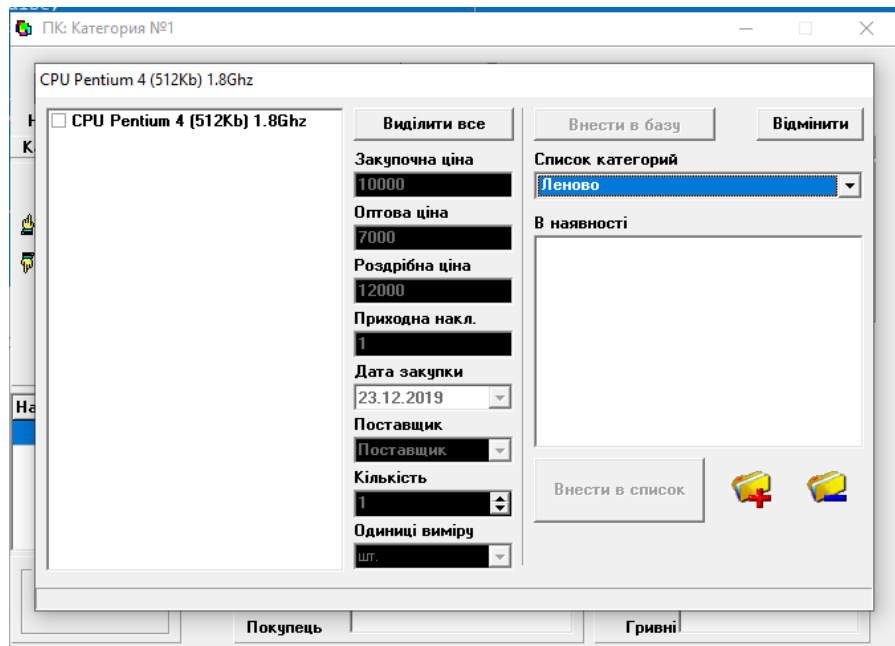


Рисунок 3.12 – Запит

Створена програма володіє рядом переваг:

1. Автоматизація необхідних розрахунків
2. Легкий спосіб введення або видалення товару або клієнта
3. Наявність у формі «Товари» кольорових зображень товару
4. Зручний дизайн програми із зрозумілим інтерфейсом і оригінальними кнопковими формами
5. Можливість перегляду або висновку на друк звітів, що наочно демонструють інформацію по основних категоріях.

Таким чином, створено підсистему, яка вирішила наступні завдання:

- облік товарів на складі;
- можливість зміни номенклатури товару;
- відображення асортименту товарів;
- відображення контрагентів;
- облік операцій з товарами по контрагентам;
- друк необхідних звітних документів.

ВИСНОВОК

В даний час на ринку програмного забезпечення існує безліч різних комерційних програмних продуктів, що дозволяють вести облік товарів на підприємстві. Такі програмні продукти не універсальні, неможливо знайти такий, який би повністю задовольняв замовника. Облік на кожному підприємстві відрізняється своєю специфікою. А переробка готового комерційного продукту під конкретне підприємство обійдеться набагато дорожче, ніж написання нового. На даний момент існує безліч програмних пакетів, за допомогою яких можливо розробляти і створювати бази даних.

В ході кваліфікаційної магістерської роботи було розроблено автоматизовану систему інформаційної підтримки клієнтів. Для досягнення поставленої мети було виконано такі задачі:

- 1) проведено аналіз предметної області,
- 2) розроблено концептуальну модель даних АІС,
- 3) розроблено логічну модель даних АІС,
- 4) розроблено інфологічну модель даних АІС,
- 5) визначено архітектуру АІС,
- 6) виконано програмну реалізацію та тестування АІС.

СПИСОК ЛІТЕРАТУРИ

1. Теорія прогнозування і прийняття рішень. Навчальний посібник. / Под ред. С. А. Саркіяна. - М.: «Вища школа», 1977. - 351 с.
2. Чуєв Ю. В., Спехова Г. П. Технічні завдання дослідження операцій. - М.: «Радянське радіо», 1971. - 244 с.
3. Чуєв Ю. В., Михайлов Ю. Б., Кузьмін В. І. Прогнозування кількісних характеристик процесів. - М.: «Радянське радіо», 1975. - 398 с.
4. Статистика. Підручник. / Под ред. проф. І. І. Єлісеєвої. - М.: ВІТРЕМ, 2002. - 448 с.
5. Сорокін А. В. Delphi. Розробка баз даних. - СПб: Пітер, 2005. - 480 с.
6. Фаронов В. В. Delphi 2005. Розробка додатків для баз даних і Інтернету. - СПб: Пітер, 2005. - 608 с.
7. Епанешников А. М., Епанешников В. А. Delphi. Програмування СУБД. - М.: ДІАЛОГ-МІФІ, 2001. - 528 с.
8. Фленов М. Е. Біблія Delphi. - СПб: ВHV, 2004. - 880 с.
9. Галісеєв Г. В. Програмування в середовищі Delphi 7. Самовчитель. - К.: Діалектика, 2003. - 288 с.
10. Керман М. К. Програмування та налагодження в Delphi. Навчальний курс. - М.: Вільямс, 2004. - 720 с.
11. Хансен Г., Хансен Дж. Бази даних: розробка і управління. - М.: БІНОМ, 1999. - 704 с.
12. Коннолі Т., Бегг К., Страчан А. Бази даних: проектування, реалізація і супровід. - М.: Вільямс, 2000. - 1120 с.
13. Праг К. Н., Ірвін М. Р. Access 2000. Біблія користувача. - М.: Вільямс, 2000. - 1040 с.
14. Єлманова Н. Системи управління базами даних провідних виробників. Комп'ютерПресс, липень 2005.
15. Гамільтон Б. ADO Збірник рецептів. Для професіоналів. Microsoft ADO.NET

16. Паронжанов С. Об'єктно-орієнтовані засоби аналізу, проектування і реінженірінга інформаційних систем. - М.: Навчальні матеріали конференції «Індустрія програмування 2008». 2008 р. – с.117-123.
17. Міжнародні стандарти, що підтримують життєвий цикл програмних засобів. М., МП «Економіка», 2007
18. Strange K. Magic Quadrant for Data Warehouse DBMSs // Gartner Research Note M-22-2154. 2008
19. Кодд Е.Ф. Реляційне уявлення даних в сховищах даних. (A Relational Model of Data for Large Shared Data Banks)
20. Соломон Д., Руссинович М. Внутрішній устрій Microsoft Windows 2000. Майстер-клас / Пер. з англ. - СПб.: Пітер; М.: Видавничо-торговий дім «Російська Редакція». 2008. - 746 стор.: ил.
21. <http://Microsoft.com/FoxPro/> - Інтернет сайт СУБД Microsoft FoxPro.
22. www.Microsoft.com Інтернет сайт компанії Microsoft.
23. www.mysql.com Інтернет сайт СУБД MySQL.
24. Мишенин, А.І. Теорія економічних інформаційних систем: Учеб. для вузів / О.І. Мішенін.- 4-е изд., Доп. і перераб. -М.: Фінанси і статистика, 2007. - 240 с.: Ил.
25. Смирнова, Г.Н. Проектування економічних інформаційних систем: Учеб. для вузів / Г.М. Смирнова, А.А. Сорокін, Ю.Ф. Тельнов; Під ред. Ю.Ф. Тельнова. -М.: Фінанси і статистика, 2006. - 512 с.: Ил.
26. Каратигін С. Бази даних: просто средства опрацювання інформації. Системи управління базами даних. М.: АБФ 2009.
27. Вендров А.М. CASE-технології. Сучасні методи і засоби проектування інформаційних систем.– М.: Фінанси і статистика, 2008
28. Дейт К. Введення в системи баз даних / Пер. з англ. М.: Наука, 2004. 463 с.
29. Архангельський А.Я. Програмування в Delphi7 М.: Біном, 2007.
30. Оскерко В.С., Пунчік З.В. Практикум з технологій баз даних. - Мн.: "БДЕУ", 2007. - 170 с.

31. Ірвін М., Праг К. Access 2000. Біблія користувача. - М.: "Діалектика", 2007. - 1040 с.
32. Microsoft Excel 2000. Видавництво "Еком", 2007 р
33. Microsoft Word 2008. Видавництво "Еком", 2008 р
34. Паронжанов С. Об'єктно-орієнтовані засоби аналізу, проектування і реінженірінга інформаційних систем. - М.: Навчальні матеріали конференції «Індустрія програмування 2006». 2006 р с.117.
35. Осипов А., Бази даних і Delphi. Теорія і практика Бази даних і Delphi. Теорія і практика Видавництво: БХВ-Петербург: 2018.
36. Понамарьов, В. Бази даних в Delphi 7. Самовчитель; СПб: Пітер - М., 2015. - 224 с.
37. Прайс, Джейсон Oracle Database 11g: SQL. Оператори SQL і програми PL / SQL / Джейсон Прайс. - М.: ЛОРИ, 2016. - 660 с.
38. Пржиялковський, В. В. Введення в Oracle SQL. - М.: Біном. Лабораторія знань, Інтернет-університет інформаційних технологій, 2013. - 320 с.
39. Прибув, Б. Oracle PL / SQL. Для професіоналів / Білл Прибув. - М.: Пітер, 2014. - 725 с.
40. Проектування і реалізація баз даних Microsoft SQL Server 2000. Навчальний курс MCSE (+ CD-ROM). - М.: Російська Редакція, 2013. - 664 с.
41. Саураб, Г. Oracle PL / SQL. Керівництво для розробників / Гупта Саураб. - М.: ЛОРИ, 2014. - 978 с.
42. Сван, Т. Секрети 32-розрядної програмування в Delphi (+ дискета); Діалектика - М., 2015. - 480 с.
43. Селко, Д. SQL для професіоналів. Програмування / Джо Селко. - М.: ЛОРИ, 2015. - 464 с.
44. Сухарєв, М.В. Основи Delphi. Професійний підхід; Наука і техніка - М., 2018. - 600 с.
45. Федоров, А. Delphi 2.0 для всіх; Комп'ютер-прес - М., 2013. - 464 с.

- 46.Форту, Б. Освой самостійно SQL за 10 хвилин / Бен Форту. - М .: Вільямс, 2015. - 288 с.
- 47.Хардман, Р. Oracle Database PL / SQL. Рекомендації експерта / Рон Хардман, М. МакЛафлін. - М .: ЛОРИ, 2014. - 450 с.

ДОДАТОК

лістинг програми

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDbClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;
using System.Globalization;
using System.IO;
using System.Text.RegularExpressions;

namespace GLOBAL2
{
    public partial class frmEditInDocs : Form
    {
        internal string rights;
        string PathToFile;
        frmEditInDocsSettings FrmEditInDocsSettings; // форма настройки
        public int limitedID { get; set; } // ConstID - айди, що обмежує відкриття одного і того ж елемента двічі
        // (наприклад при редагуванні)
        public int rowStatus { get; set; } // статус рядки, (-1) - режим вибору:
        кліком по ґрид
        public bool isSDI { get; set; }
        public event EventHandler<EventArgs> RaiseInDocSelected;
        DataTable tblInDocs = new DataTable("InDocs");
        DataTable tblInDocsRow = new DataTable("InDocsRow");
        DataTable tblImageList = new DataTable("ImageList");
        DataSet ds = new DataSet();
        ImageList imgList = new ImageList();
        CurrentFilter currentFilter;
        bool checkChanges;
        bool exitCancel;
        DataRow row;
        byte[] Doc;
        int dgvRowIndex;
        int tblRowIndex;
        int currentConstID;
        bool EntryInDgv = false;
        DateTime DefaultDate { get; set; }
        DateTime DefaultInDate { get; set; }
        int DefaultInDocType { get; set; }
        public byte[] Doc
        {
            get { return doc; }
            set { Doc = value; }
        }
        public frmEditInDocs ()
        {
            InitializeComponent ();
            dgvInDocs.AutoGenerateColumns = false;
            imgList.Images.Add (GLOBAL2.Properties.Resources.document);
        }
        // завантаження форми з заявками

```

```

        private void frmEditInDocs_Load (object sender, EventArgs e)
        {
            ComboBoxFill (cboDept, @ "SELECT ID,
            Description
            FROM dbo.Dept AS d
            ORDER BY
            Description ");
            ComboBoxFill (cboInDocType, "SELECT ID, Description, NameInFile FROM InDocTypes Order By
            Description");

            ds.Tables.Add (tblInDocs);
            ds.Tables.Add (tblInDocsRow);
            ds.Tables.Add (tblImageList);
            checkChanges = false;
            DgvInDocsFill ();
            dgvInDocs.DataSource = tblInDocs;
            UpdateHeaderText ();
            if (! FillTable (tblImageList, "SELECT * FROM ImageList ORDER BY
            GroupName, Ord")) {Close (); }
            SwitchMode (rowStatus);
            this.Location =
            GLOBAL2.Properties.Settings.Default.frmEditInDocsLocation;
            DefaultDate = DateTime.Now;
            DefaultInDate = DateTime.Now;
            DefaultInDocType = 0;
        }

        //
        //
        // заповнення комбо-боксу з клієнтами
        private void ComboBoxFill (ComboBox cb, string cmdtext)
        {
            OleDbCommand cmd = new OleDbCommand (cmdtext, new
            OleDbConnection (SessionProperties.currentConnectionString));
            DataTable dt = new DataTable ();
            try
            {
                cmd.Connection.Open ();
                dt.Load (cmd.ExecuteReader ());
                cb.DataSource = dt;
            }
            catch (Exception ex)
            {
                MessageBox.Show (ex.Message, "Помилка !!!", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
            finally
            {
                cmd.Connection.Close ();
            }
            cb.ValueMember = "ID";
            cb.DisplayMember = "Description";
        }

        // заповнення таблиць
        private bool FillTable (DataTable tbl, string OleDbText)
        {
            OleDbConnection cn = new
            OleDbConnection (SessionProperties.currentConnectionString);
            OleDbCommand cmd = new OleDbCommand ();
            cmd.Connection = cn;
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = OleDbText;
            OleDbDataReader reader = null;
            tbl.Clear ();

```

```

tbl.Columns.Clear ();
try
{
    cn.Open ();
    reader = cmd.ExecuteReader ();
    tbl.Load (reader);
    cn.Close ();
}
catch (Exception ex)
{
    MessageBox.Show (ex.Message, "FillTable (" + Tbl.ToString () +
", " + OleDbText + ")", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    return false;
}

return true;
}

} // малювання картинок на контроль * /

private void SwitchMode (int mode)
{
    switch (Mode)
    {
        case -1: // режим вибору
            tsbPeriod.Enabled = true;
            tsbDocNew.Enabled = false;
            tsbDocEdit.Enabled = false;
            tsbDocDelete.Enabled = false;

            tsbHistory.Enabled = false;
            tsbSettings.Enabled = true;

            cboDept.Enabled = false;
            cboInDocType.Enabled = false;
            tbNum.Enabled = false;
            dtpDate.Enabled = false;
            dtpInDate.Enabled = false;
            tbDescription.Enabled = false;
            btnImgFile.Enabled = false;
            btnOk.Enabled = false;

            btnScan.Enabled = false;
            exitCancel = true;

            dgvInDocs.Enabled = true;

            dgvInDocs.Focus ();
            break;
        case 0://новий
            tsbPeriod.Enabled = false;
            tsbDocNew.Enabled = false;
            tsbDocEdit.Enabled = false;
            tsbDocDelete.Enabled = false;

            tsbSettings.Enabled = false;

            cboDept.SelectedValue = 0;
            cboDept.Enabled = true;
            cboInDocType.SelectedValue = DefaultInDocType;
            cboInDocType.Enabled = true;
            tbNum.Text = "";
            tbNum.Enabled = true;

            dtpDate.Value = DefaultDate;

            dtpDate.Enabled = true;

            dtpInDate.Value = DefaultInDate;

            dtpInDate.Enabled = true;
            tbDescription.Text = "";
            tbDescription.Enabled = true;

```

```

        lblImgFile.Text = "";
        btnImgFile.Enabled = true;

        btnFolder.Enabled = true;
        btnScan.Enabled = true;

        btnOk.Enabled = false;
        exitCancel = false;
        checkChanges = true;
        rowStatus = mode;
        btnExit.Text = "Esc - відміна";
        EntryInDgv = false;

        dgvInDocs.Enabled = false;

        cboDept.Focus ();
        break;
        case 1:// правка
            tsbPeriod.Enabled = false;
            tsbDocNew.Enabled = false;
            tsbDocEdit.Enabled = false;
            tsbDocDelete.Enabled = false;

            tsbSettings.Enabled = false;

            cboDept.Enabled = true;
            cboInDocType.Enabled = true;
            tbNum.Enabled = true;
            dtpDate.Enabled = true;
            dtpInDate.Enabled = true;
            tbDescription.Enabled = true;
            btnImgFile.Enabled = true;

            btnFolder.Enabled = true;
            btnScan.Enabled = true;
            btnOk.Enabled = false;

            //dgvInDocs.Enabled = false;
            exitCancel = false;
            checkChanges = true;
            rowStatus = mode;
            btnExit.Text = "Esc - відміна";
            EntryInDgv = false;

            dgvInDocs.Enabled = false;

            cboDept.Focus ();
            break;
            case 2:// перегляд документів
                tsbPeriod.Enabled = true;
                tsbDocNew.Enabled = true;
                tsbDocEdit.Enabled = true;
                tsbDocDelete.Enabled = true;

                tsbSettings.Enabled = true;

                cboDept.Enabled = false;

                cboDept.SelectedValue = 0;
                cboInDocType.Enabled = false;
                cboInDocType.SelectedValue = 0;

                tbNum.Enabled = false;
                dtpDate.Enabled = false;
                dtpInDate.Enabled = false;
                tbDescription.Enabled = false;
                btnImgFile.Enabled = false;

            btnFolder.Enabled = false;

            btnOk.Enabled = false;

            dgvInDocs.Enabled = true;
            exitCancel = true;
            checkChanges = false;
            btnExit.Text = "Esc - вихід";
            if (this.Text.Contains ("*"))
            {

```

```

        this.Text = this.Text.Substring (0,
this.Text.Length - 1);
    }
    dgvInDocs.Enabled = true;
        dgvInDocs.Focus ();
        break;
    }
    }// установка режиму форми

private void ControlsRefresh (DataGridViewCellEventArgs e)
{
    if (! EntryInDgv)
    {
        SwitchMode (2);
        EntryInDgv = true;
    }
    row = ((DataRowView) DgvInDocs.Rows [e.RowIndex] .DataBoundItem)
.Row;

    tblRowIndex = Array.IndexOf (tblInDocs.Select (), row);
    cboDept.SelectedValue = row ["DeptID"] .ToString ();
    cboDept.Text = row ["Dept"] .ToString ();
    cboInDocType.SelectedValue = row ["InDocTypesID"] .ToString ();
    cboInDocType.Text = row ["DocType"] .ToString ();
    tbNum.Text = row ["Num"] .ToString ();
    dtpDate.Value = Convert.ToDateTime (row ["Date"]);
    dtpInDate.Value = Convert.ToDateTime (row ["InDate"]);
    tbDescription.Text = row ["Description"] .ToString ();
    lblImgFile.Text = row ["ImgFile"] .ToString ();
    currentConstID = (int) Row ["ConstID"];

    }// оновлення пов'язаних з поточним рядком контроль

public static bool SaveChanges (DateTime Date, DateTime InDate,
int ConstID = 0,
int InDocTypeID = 0,
string Num = "",
int DeptID = 0,
string Description = "",
string ImgFile = "",
byte[] Img = null,
int rowStatus = 0)
{
    OleDbConnection cn = new
OleDbConnection(SessionProperties.currentConnectionString);
    OleDbCommand cmd = new OleDbCommand("Dbo.EditInDocs", Cn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandTimeout = 0;
    cmd.Parameters.Add ("@ConstID", OleDbType.Int);
    cmd.Parameters ["@ConstID"] .Value = ConstID;

    cmd.Parameters.Add ("@InDocTypesID", OleDbType.Int);
    cmd.Parameters ["@InDocTypesID"] .Value = InDocTypeID;

    cmd.Parameters.Add ("@Num", OleDbType.NVarChar, 20);
    cmd.Parameters ["@Num"] .Value = Num;

    cmd.Parameters.Add ("@Date", OleDbType.DateTime);
    cmd.Parameters ["@Date"] .Value = Date;

    cmd.Parameters.Add ("@DeptID", OleDbType.Int);
    cmd.Parameters ["@DeptID"] .Value = DeptID;

    cmd.Parameters.Add ("@InDate", OleDbType.DateTime);

```

```

cmd.Parameters ["@InDate"] .Value = InDate;

cmd.Parameters.Add ("@Description", OleDbType.NVarChar, 254);
cmd.Parameters ["@Description"] .Value = Description;

cmd.Parameters.Add ("@ImgFile", OleDbType.NVarChar, 254);
cmd.Parameters ["@ImgFile"] .Value = ImgFile;

cmd.Parameters.Add ("@Img", OleDbType.Image);
cmd.Parameters ["@Img"] .Value = Img;

cmd.Parameters.Add ("@SessionsID", OleDbType.Int);
cmd.Parameters ["@SessionsID"] .Value = SessionProperties.sessionID;
cmd.Parameters.Add ("@RowStatus", OleDbType.Int);
cmd.Parameters ["@RowStatus"] .Value = rowStatus;
cmd.Parameters.Add ("@Msg", OleDbType.NVarChar, 400);
cmd.Parameters ["@Msg"] .Value = "";
cmd.Parameters ["@Msg"] .Direction = ParameterDirection.Output;
cmd.Parameters.Add ("@Res", OleDbType.Int);
cmd.Parameters ["@Res"] .Value = 0;
cmd.Parameters ["@Res"] .Direction = ParameterDirection.Output;
SessionProperties.lastError = "";
try
{
    cn.Open ();
    cmd.ExecuteNonQuery ();
    cn.Close ();
    int result = (int) Cmd.Parameters ["@Res"] .Value;
    if (Result >= 0)
    {
        return true; // успіх
    }
    else
    {
        MessageBox.Show ((string) Cmd.Parameters ["@Msg"]
.Value, "Помилка !!!", MessageBoxButtons.OK, MessageBoxIcon.Stop);
        return false; // Помилка збереженої процедури
    }
}
catch (Exception ex)
{
    SessionProperties.lastError = ex.Message;
    MessageBox.Show (SessionProperties.lastError, "Помилка !!!",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
    return false; // Помилка OLEDB
}
} // збереження змін

private void DataChanged (object sender, EventArgs e)
{
    if (CheckChanges)
    {
        if (this.Text.IndexOf ('*') == -1)
        {
            this.Text += "*";
        }
        btnOk.Enabled = true;
    }
}
// вивід заявок в ґрид за певний період
private void DgvInDocsFill ()
{

```

```

OleDbCommand cmd = new OleDbCommand("SELECT * FROM PCOrders WHERE Date BETWEEN @begDate
AND @endDate ORDER BY Description", new
OleDbConnection(SessionProperties.currentConnectionString));
cmd.Parameters.Add ("@BegDate",OleDbDbType.DateTime);
cmd.Parameters.Add ("@EndDate",OleDbDbType.DateTime);
cmd.Parameters ["@BegDate"] .Value = SessionProperties.begDate;
cmd.Parameters ["@EndDate"] .Value = SessionProperties.endDate;

try
{
cmd.Connection.Open ();
tblInDocs.Clear ();
tblInDocs.Load (cmd.ExecuteReader ());
}
catch (Exception ex)
{
MessageBox.Show (ex.Message, "DgvInDocsFill ()",MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
finally
{
cmd.Connection.Close ();
}

//зміна заголовка
private void UpdateHeaderText ()
{
this.Text = "Заявки за період з" +
SessionProperties.begDate.ToShortDateString () + "По" +
SessionProperties.endDate.ToShortDateString ();
}

// видалення заявки
private void DocDelete ()
{
var result = MessageBox.Show ("Видалити документ?", "Увага???",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
if (Result == DialogResult.Yes)
{
if (SaveChanges (Date: dtpDate.Value, InDate: dtpInDate.Value, ConstID: currentConstID,
rowStatus: 2))
{
DgvInDocsFill ();
RestoreRowIndexPosition ();
}
}
}

// вихід
private void btnExit_Click (object sender, EventArgs e)
{
if (ExitCancel)
{
Close ();
}
else
{
btnExit.Text = "Esc - Вихід";
SwitchMode (2);
}
}

//збереження нової заявки
private void btnOk_Click (object sender, EventArgs e)

```

```

    {
        if (RowStatus == 0)
        {
            DataTable dt = SearchSimilarDoc ((int) CboDept.SelectedValue, (int)
CboInDocType.SelectedValue, tbNum.Text);
            if (Dt.Rows.Count > 0)
            {
                frmEditInDocsChoosingReplaceDoc _frmEditInDocsChoosingReplaceDoc = new
frmEditInDocsChoosingReplaceDoc (Dt);
                DialogResult dResult = _frmEditInDocsChoosingReplaceDoc.ShowDialog ();
                if (DialogResult.Yes == dResult)
                {
                    rowStatus = 1;
                    currentConstID = SessionProperties.retInt;
                }
                if (DialogResult.Cancel == dResult)
                {
                    return;
                }
            }
            if (LblImgFile.Text == "" && doc == null)
            {
                MessageBox.Show ("Немає документа !!!", "Увага!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                return;
            }
            if (SaveChanges (dtpDate.Value, dtpInDate.Value, InDocTypeID: (int)
CboInDocType.SelectedValue, ConstID: currentConstID,
Num: tbNum.Text, DeptID: (int) CboDept.SelectedValue, Description: tbDescription.Text,
ImgFile: lblImgFile.Text, Img: doc, rowStatus: rowStatus))
            {
                DefaultDate = dtpDate.Value;
                DefaultInDate = dtpInDate.Value;
                DefaultInDocType = (int) CboInDocType.SelectedValue;
                DgvInDocsFill ();

                RestoreRowIndexPosition ();

                doc = null;

                SwitchMode (2);
            }
        }
        //пошук заявки
        private DataTable SearchSimilarDoc (int DeptID, int InDocTypesID, string DocNum)
        {
            DataTable resTable = new DataTable ();
            OleDbCommand cmd = new OleDbCommand ("SELECT * FROM PCOrders WHERE DeptID = @DeptID AND
InDocTypesID = @InDocTypesID AND Num = @DocNum ORDER BY Description", new
OleDbConnection (SessionProperties.currentConnectionString));
            cmd.Parameters.Add ("@DeptID", OleDbType.Int) .Value = DeptID;
            cmd.Parameters.Add ("@InDocTypesID", OleDbType.Int) .Value = InDocTypesID;
            cmd.Parameters.Add ("@DocNum", OleDbType.NVarChar) .Value = DocNum;
            OleDbDataAdapter da = new OleDbDataAdapter (cmd);
            da.Fill (resTable);
            return resTable;
        }

        private void RestoreRowIndexPosition ()
        {
            try
            {
                if (DgvInDocs.Rows.Count > 0)
                {

```



```

DataRow[] SearchedRows = ((DataTable) DgvInDocs.DataSource) .Select ("ConstID =" +
CurrentConstID.ToString ());
int rowIndex;
if (SearchedRows.Length> 0)
{
rowIndex = tblInDocs.Rows.IndexOf (searchedRows [0]);
}
else
{
rowIndex = (dgvRowIndex> = dgvInDocs.Rows.Count)? dgvInDocs.Rows.Count - 1: dgvRowIndex;
}
if (RowIndex> dgvInDocs.Rows.Count - 1)
{
dgvInDocs.CurrentCell = dgvInDocs [0, 0];
}
else
dgvInDocs.CurrentCell = dgvInDocs [0, rowIndex];
}
}
catch (Exception ex)
{
MessageBox.Show (ex.Message, "RestoreRowIndexPosition
()", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void dgvInDocs_CellMouseDoubleClick (object sender,
DataGridViewCellMouseEventArgs e)
{
if (RowStatus == -1)// режим вибору
{
DataRow row = ((DataRowView) DgvInDocs.Rows [e.RowIndex]
.DataBoundItem) .Row;// рядок з прив'язаною таблиці
SessionProperties.retInt = (int) Row ["ConstID");// повернення
(збереження в статичному об'єкті) значення з форми
SessionProperties.retString = (string) Row ["FullName"];
EventHandler<EventArgs> Handler = RaiseInDocSelected;

if (Handler! = null)
{
// оператор для виклику події.
handler (this, E);
}
Close ();
return;
}
if (E.RowIndex> = 0)
{
dgvRowIndex = dgvInDocs.CurrentRow.Index;
rowStatus = 1;
tsbDocEdit_Click (null, null);
// фільтр
}
}
//оновити список заявок
private void dgvInDocs_RowEnter (object sender, DataGridViewCellEventArgs e)
{
ControlsRefresh (e);
}
// права на створення заявки
private void tsbDocNew_Click (object sender, EventArgs e)
{

```

```

if (Rights.Contains ("A"))
SwitchMode (0);
else
{
    MessageBox.Show ("У вас немає прав на додавання документів !!!", "Зверніться до адміністратора !!!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
// права на редагування заявки
private void tsbDocEdit_Click (object sender, EventArgs e)
{
if (Rights.Contains ("E"))
{
    SwitchMode (1);
}
else
{
    MessageBox.Show ("У вас немає прав на редагування документів !!!", "Зверніться до адміністратора !!!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
//звітна форма
private void btnImage_Click (object sender, EventArgs e)
{
    OpenFileDialog ofdInDoc = new OpenFileDialog();
    string val = lblImgFile.Text;
    if (Val == "")
    {
        val = GetDefaultSetting.Get ("DefaultInDocImgPath");
        if (Val == "") { return; }
    }
    ofdInDoc.InitialDirectory = val;
    val = GetDefaultSetting.Get ("DefaultInDocImgFilter");
    if (Val == "") { return; }
    ofdInDoc.Filter = val;
    if (OfdInDoc.ShowDialog () == DialogResult.OK)
    {
        lblImgFile.Text = ofdInDoc.FileName;
        FillValueAfterOpenFileDialog (ofdInDoc.FileName);
    }
    Environment.CurrentDirectory = Application.StartupPath;
}
//установка налаштувань
private void tsbSetting_Click (object sender, EventArgs e)
{
    frmPeriodSelector fps = new frmPeriodSelector();
    //fps.Owner = this;
    //fps.MdiParent = this.MdiParent;
    if (Fps.ShowDialog () == DialogResult.OK)
    {
        DgvInDocsFill ();
        UpdateHeaderText ();
    }
}
// права на видалення заявок
private void tsbDocDelete_Click (object sender, EventArgs e)
{
if (Rights.Contains ("D"))
    DocDelete ();
else
{

```

```

    MessageBox.Show ("У вас немає прав на видалення документів !!!", "Зверніться до адміністратора !!!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
// Натискання кнопки на клавіатурі по datagridview
private void dgvInDocs_KeyDown (object sender, KeyEventArgs e)
{
    switch (E.KeyCode)
    {
        case Keys.Enter:
            e.SuppressKeyPress = true;
            if (DgvInDocs.CurrentRow! = null)
            {
                DataRow row = ((DataRowView)
DgvInDocs.CurrentRow.DataBoundItem) .Row;
                SwitchMode (1);
            }
            break;
        case Keys.Insert:
            SwitchMode (0);
            break;
        case Keys.Delete:
            dgvRowIndex = dgvInDocs.CurrentRow.Index;
            DocDelete ();
            break;
    }
    if (E.Control && Keys.O == e.KeyCode)
    {
        if (DgvInDocs.CurrentRow! = null)
            OpenDoc ((int) DgvInDocs.CurrentRow.Cells ["ID"] .Value, (int)
DgvInDocs.CurrentRow.Cells ["ConstID"] .Value);
    }
}
//нова заявка
private void tsmiDocNew_Click (object sender, EventArgs e)
{
    // SwitchMode (0);
    tsbDocNew_Click (null, null);
}
//редагування завком
private void tsmiDocEdit_Click (object sender, EventArgs e)
{
    // SwitchMode (1);
    tsbDocEdit_Click (null, null);
}
//видалення заявки
private void tsmiDocDelete_Click (object sender, EventArgs e)
{
    tsbDocDelete_Click (null, null);
    // DocDelete ();
}
//збереження після редагування
private void frmEditInDocs_FormClosed (object sender, FormClosedEventArgs e)
{
    GLOBAL2.Properties.Settings.Default.frmEditInDocsLocation =
this.Location;
    GLOBAL2.Properties.Settings.Default.Save ();
}

// Форма настройки поиска
private void tsbSettings_Click (object sender, EventArgs e)
{
    if (FrmEditInDocsSettings! = null)

```

```

if (FrmEditInDocsSettings.IsDisposed == true)
FrmEditInDocsSettings = null;
if (FrmEditInDocsSettings == null)
{

FrmEditInDocsSettings = new frmEditInDocsSettings();
FrmEditInDocsSettings.invokeSearch = Search;
FrmEditInDocsSettings.Owner = this.Owner;
FrmEditInDocsSettings.MdiParent = this.MdiParent;
FrmEditInDocsSettings.dtDept = ((DataTable) CboDept.DataSource) .Copy ();
FrmEditInDocsSettings.dtTypeDoc = ((DataTable) CboInDocType.DataSource) .Copy ();
FrmEditInDocsSettings.frmEIDs = this;
FrmEditInDocsSettings.Show ();
if (CurrentFilter! = null)
{
FrmEditInDocsSettings.SetControlValue (currentFilter.Dept, currentFilter.TypeDoc,
currentFilter.NumDoc,
currentFilter.InNumDoc, currentFilter.DateDoc, currentFilter.InDateDoc,
currentFilter.Description);
}
}
else
FrmEditInDocsSettings.Focus ();
}

// Натискання правої кнопки миші по осередку datagridview
private void dgvInDocs_CellMouseClick (object sender, DataGridViewCellMouseEventArgs e)
{
if (E.ColumnIndex >= 0 && e.RowIndex >= 0)
{
if (E.Button == MouseButtons.Right)
{
PathToFile = dgvInDocs.Rows [e.RowIndex] .Cells ["FileName"] .Value.ToString ();
dgvInDocs.Rows [e.RowIndex] .Cells [e.ColumnIndex] .ContextMenuStrip = cmsInDocs; //
cmsInDocs;
cmsInDocs.Show (MousePosition);
dgvInDocs.Rows [e.RowIndex] .Cells [e.ColumnIndex] .ContextMenuStrip = null;
tsmiOpenDoc.Tag = new int[] {(int) DgvInDocs.Rows [e.RowIndex] .Cells ["ID"] .Value,
(int) DgvInDocs.Rows [e.RowIndex] .Cells ["ConstID"] .Value};
}
}
}
//
// Обробка натискання "Відкрити заявку"
//
private void tsmiOpenDoc_Click (object sender, EventArgs e)
{
ToolStripMenuItem tsmi = (ToolStripMenuItem) Sender;
if (Tsmi.Tag! = null)
{
int[] Array = (int[]) Tsmi.Tag;

// Прочитати документ з бази
OpenDoc (array [0]/ * ID * /, Array [1]/ * ConstID * /);
}
}
//
//
// Завантаження документа
private static byte[] LoadImg (int ID, int ConstID)
{
OleDbCommand cmd = new OleDbCommand();
cmd.Connection = new OleDbConnection(SessionProperties.currentConnectionString);

```

```

cmd.CommandText = "SELECT Img FROM InDocs WHERE Img is not null and ConstID = @ ConstID
and ID = @ID";
cmd.CommandTimeout = 0;

cmd.Parameters.Add ("@ID", OleDbType.Int);
cmd.Parameters ["@ID"] .Value = ID;

cmd.Parameters.Add ("@ConstID", OleDbType.Int);
cmd.Parameters ["@ConstID"] .Value = ConstID;

OleDbDataAdapter da = new OleDbDataAdapter(Cmd);
DataTable dt = new DataTable();
da.Fill (dt);
if (Dt.Rows.Count == 0)
{
    MessageBox.Show ("Немає документа в базі даних");
    return null;
}
else
return (byte[]) Dt.Rows [0] ["Img"];
}
// відкриття документа
public static bool OpenDoc (int ID, int ConstID)
{
    byte[] FileData = LoadImg (ID, ConstID);
    if (FileData! = null)
    {
        string fileName = @ "\\ Docx \ report \" + DateTime.Now.ToFileTime (). ToString () +
        ".Pdf";
        File.WriteAllBytes (fileName, fileData);
        System.Diagnostics.Process.Start (fileName);
        return true;
    }
    else return false;
}
// Фільтр відображаються документів
private void Search (string Dept = "", string TypeDoc = "", string NumDoc = "", string
InNumDoc = "", string DateDoc = "", string InDateDoc = "", string Description = "", bool
ClearRowFilter = false)
{
    if (ClearRowFilter)
    {
        ((DataTable) (DgvInDocs.DataSource)). DefaultView.RowFilter = "";
        currentFilter = null;
        return;
    }
    DataTable dt = new DataTable();
    dt = (DataTable) (DgvInDocs.DataSource);
    dt.DefaultView.RowFilter = "";

    currentFilter = new CurrentFilter(Dept, TypeDoc, NumDoc,
    InNumDoc, DateDoc, InDateDoc, Description);

    if (Dept! = "" && dgvInDocs.RowCount> 0)
    dt.DefaultView.RowFilter = "Dept =" + " '" + Dept + " '";

    if (TypeDoc! = "" && dgvInDocs.RowCount> 0)
    dt.DefaultView.RowFilter += (dt.DefaultView.RowFilter == "" ? "" : "And" ) + "DocType ="
+ " '" + TypeDoc + " '";

    if (NumDoc! = "" && dgvInDocs.RowCount> 0)
    dt.DefaultView.RowFilter += (dt.DefaultView.RowFilter == "" ? "" : "And" ) + "Num
Like"+" '%" + NumDoc +"% '";

```

```

if (InNumDoc! = "" && dgvInDocs.RowCount > 0)
    dt.DefaultView.RowFilter += (dt.DefaultView.RowFilter == "" ? "" : "And") + "ConstID ="
+ int.Parse (InNumDoc);

if (DateDoc! = null && dgvInDocs.RowCount > 0)
    dt.DefaultView.RowFilter += (dt.DefaultView.RowFilter == "" ? "" : "And") + "Convert
(Date, 'System.String') Like '" + DateDoc + "% '";

if (InDateDoc! = null && dgvInDocs.RowCount > 0)
    dt.DefaultView.RowFilter += (dt.DefaultView.RowFilter == "" ? "" : "And") + "Convert
(InDate, 'System.String') Like '" + InDateDoc + "% '";

if (Description! = "")
    dt.DefaultView.RowFilter += (dt.DefaultView.RowFilter == "" ? "" : "And") +
"Description Like '%" + Description + "% '";
}

// Заміна документа
private void btnScan_Click (object sender, EventArgs e)
{
    // if (ApplicationManager.flagScannerInstalled == true)
    // {
        if (RowStatus == 0)
        {
            Show_frmEditInDocsScan ();
        }
        else if (RowStatus == 1)
        {
            System.Windows.Forms.DialogResult dlgResult =
MessageBox.Show ("Замінити поточний документ? \ N \ " Так \ "- замінити; \ n \ " Ні \ "-
відкрити на редагування.", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (DlgResult == System.Windows.Forms.DialogResult.Yes)
            {
                Show_frmEditInDocsScan ();
            }
            else if (DlgResult ==
System.Windows.Forms.DialogResult.No)
            {
                List<Image> Imgs = null ;
                doc = frmEditInDocs.LoadImg ((int)
DgvInDocs.CurrentRow.Cells ["ID"] .Value, (int) DgvInDocs.CurrentRow.Cells ["ConstID"]
.Value);

                try
                {
                    imgs = ExtractImages (doc);
                }
                catch (Exception ex)
                {
                    MessageBox.Show (ex.Message, "Помилка
!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    return;
                }
                if (Imgs! = null)
                {
                    Show_frmEditInDocsScan (imgs);
                }
            }
        }
    }
}

```