

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Розробка комп'ютеризованої системи управління формуванням
доставкою замовлень»**

Студентки 4 курсу, 13 групи,
спеціальності
122 «Комп'ютерні науки»

Устименко
Назар
Ігорович

підпис студента

Науковий керівник
кандидат фізико-математичних наук

Філімонова Тетяна
Олегівна

підпис керівника

Гарант освітньої програми
кандидат технічних наук, доцент

Демідов Павло
Георгійович

підпис керівника

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних
Спеціальність 122 «Комп'ютерні науки»

Затверджую
Зав. кафедри _____ Пурський О.І.
«18» грудня 2021р.

Завдання на випускну кваліфікаційну роботу студенту

Устименко Назар Ігорович
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Розробка комп'ютеризованої системи управління формуванням доставкою замовлень»

Затверджена наказом ректора від «15» грудня 2020 р. № 3780

2. Строк здачі студентом закінченої роботи 31 травня 2021 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: обґрунтування та розробка комп'ютеризованої системи замовлень, з урахуванням сучасних тенденцій у сфері обслуговування.

Об'єкт дослідження: процес обробки онлайн замовлень.

Предмет дослідження: засоби створення комп'ютеризованих систем замовлень.

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Філімонова Т. О.	22.12.2020 р.	22.12.2020 р.
2	Філімонова Т. О.	22.12.2020 р.	22.12.2020 р.

3	Філімонова Т. О.	22.12.2020 р.	22.12.2020 р.
---	------------------	---------------	---------------

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Системи замовлень

1.1. Основні поняття системи замовлень

1.2. Основні принципи і переваги впровадження електронних систем замовлення

1.3. Проблеми впровадження систем впровадження електронних систем замовлення

РОЗДІЛ 2. Організація розробки комп'ютеризованої системи замовлень

2.1. Опис проблем у сфері обслуговування замовлень

2.2. Мета і призначення розробки комп'ютеризованої системи замовлень

РОЗДІЛ 3. Практична реалізація комп'ютеризованої системи замовлення

3.1. Розробка користувацького інтерфейсу

3.2. База даних та програмування системи

3.3. Оцінка отриманих результатів проектування

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Додатки

7. Календарний план виконання роботи

№ Пор .	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	15.10.2020	15.10.2020
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	18.12.2020	18.12.2020
3	<i>Вступ</i>	03.02.2020	03.02.2020
4	<i>РОЗДІЛ 1. Системи замовлень ..</i>	26.02.2020	26.02.2020
5	<i>РОЗДІЛ 2. Організація розробки комп'ютеризованої системи замовлень</i>	06.04.2020	06.04.2020
6	<i>РОЗДІЛ 3 . Практична реалізація комп'ютеризованої системи замовлення</i>	12.05.2020	12.05.2020
7	<i>Висновки</i>	14.05.2020	14.05.2020
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	20.05.2020	20.05.2020
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	26.05.2020	
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	27.05.2020	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	29.05.2020	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «22» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи (проєкту)

Філімонова Т.О.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Устименко Н.І.

(прізвище, ініціали, підпис)

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. Системи замовлень	12
1.1. Основні поняття системи замовлень	12
1.2. Основні принципи і переваги впровадження електронних систем замовлення	13
1.3. Проблеми впровадження електронних систем замовлення	23
РОЗДІЛ 2. Організація розробки комп'ютеризованої системи замовлень	26
2.1. Опис проблем у сфері обслуговування замовлень	26
2.2. Мета і призначення комп'ютеризованої системи замовлень	32
РОЗДІЛ 3. Практична реалізація комп'ютеризованої системи замовлення	38
3.1. Розробка користувацького інтерфейсу	38
3.2. База даних та програмування системи	49
3.3. Оцінка отриманих результатів проектування	58
ВИСНОВКИ	60
Список використаних джерел	62
Додаток	63

АНОТАЦІЯ

Випускна кваліфікаційна робота виконана на здобуття ступіня “Бакалавр” зі спеціальності 122 “Комп’ютерні науки”. Результатом виконання роботи є створена комп’ютеризована система замовлень, за допомогою якої можна керувати замовленнями.

Перший розділ присвячено аналізу основних понять системи замовлень. Проведено аналіз основних принципів та переваг впровадження електронних систем замовлень. Досліджені основні проблеми впровадження електронних систем замовлень.

У другому розділі описана організація розробки комп’ютеризованої системи замовлень, проведений опис проблем у сфері обслуговування замовлення з якими можна зіткнутися, та описана мета та призначення майбутньої системи.

У третьому розділі представлена програмна розробка системи. Описана покрокова технологія розробки програми та результат.

Система дозволяє управляти замовленнями, а саме: додавання, видалення, редагування записів бази даних що стосуються користувачів, товарів, послуг та замовлень доступних для користувача системи. У процесі розробки були використані мова програмування Object Pascal, СУБД Microsoft Access .

Робота складається з 72 сторінок.

Ключові слова: комп’ютеризована система замовлень, модуль управління, базові логістичні системи.

ABSTRACT

This diploma project is fulfilled for obtaining the “Bachelor” degree, 122 "Computer Science". The result of the the thes is the created computerized order system, with the help of which you can manage orders.

The first section is devoted to the analysis of the basic concepts of the ordering system. The analysis of the basic principles and advantages of introduction of electronic ordering systems is carried out. The main problems of implementation of electronic ordering systems are investigated.

The second section describes the organization of the development of a computerized ordering system, describes the problems in the field of order service that may be encountered, and describes the purpose and purpose of the future system.

The third section presents the software development of the system. Describes step-by-step program development technology and results.

The system allows you to manage orders, namely: adding, deleting, editing database records related to users, products, services and orders available to the user of the system. The Object Pascal programming language, Microsoft Access DBMS, was used in the development process.

The work consists of 58 pages.

СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ЖЕК - Житлово-експлуатаційна контора

OMS - order management system

EDI - Electronic Data Interchange

ГІС - Геоінформаційна система

GPS - Global Positioning System

ПЗ – Програмне забезпечення

CRM - Customer Relationship Management

EDI - Electronic data interchange

НДДКР - Науково-дослідні та дослідно-конструкторські роботи

ТМЦ - Товарно-матеріальні цінності

API - Application Programming Interface

VCL - Visual Component Library

FMX - FireMonkey

SQL - Structured query language

БД - База даних

ККД – Коефіцієнт корисної дії

ВСТУП

2020 рік показав Україні її майбутній шлях, по якому вона має йти. Ми вкотре впевнились що Україна має йти шляхом Європейських країн, нашій країні давно потрібно покинути цей клуб під назвою “СНГ”. Але щоб називатись по справжньому країною Європи нам потрібно відповідати цій назві.

Чим відрізняються країни Європи від України, Росії, Білорусі? Рівнем заробітної праці, чи можливо якістю життя? Обидва варіанти правильні, але як вони цього добилися? Чому скажімо Японія, острівна країна з площею вдвічі меншою ніж Україна і яка майже не має корисних копалин, завжди знаходиться на вершинах в різних топів? Всі знають правильну відповідь – це технології, які оптимізують роботу в декілька разів. Якщо українцю для виконання завдання потрібно чотири години, то японцеві в тричі менше, тобто один рік роботи японця рівносильний трьом рокам українця. Нажаль так трапалося не через державну політику України в сфері ІТ, а через населення яке не хоче адаптуватись і освоюватись в сфері нових технологій. Щоб в цьому переконатись потрібно просто прийти в вихідний день до будь-якого ЖЕК або в лікарню, ви гарантовано побачите чергу яка затягується на декілька годин. Ці черги нам дісталися з часів ССРСР, коли вся країна стояла в чергах.

На щастя сучасне покоління яке народжується “зі смартфоном в руках” цінує свій час і охоче освоює нововведення в ІТ сфері. Люди нового покоління не будуть чекати в черзі якщо це можна зробити за допомогою телефона або комп’ютера. Вони скоріше наймуть когось якщо справа стосується фізичної роботи, вони рідко готують, купують одяг сидячи на дивані і надають перевагу пасивному відпочинку. Але основною вимогою для них являється час за який вони готові щедро платити, найкращий вихід для них стане той, який займе найменше часу. Такі клієнти ”золота жила” для підприємців, але для того щоб з ними комунікувати потрібно перейти на їх мову, мову комп’ютерних технологій. Події 2020 року прискорили перехід компаній на комп’ютеризовану роботу[6]. Наприклад замовлення більшість

компаній обробляє за допомогою комп'ютеризованих систем які дозволяють оптимізувати роботу й уникнути багатьох помилок. Тому тема моєї курсової дуже актуальна на фоні сучасних подій.

Актуальність теми: пов'язана з необхідністю створення системи для управління замовленнями під працівників які не мають достатнього досвіду роботи з системою.

Мета роботи: обґрунтування та розробка комп'ютеризованої системи замовлень.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

- Проаналізувати систему обробки замовлення;
- Визначити основні переваги і недоліки встановлення системи замовлень;
- Розробити метод отримання і обробки замовлень на складі.
- Дослідити інструментальні засоби, що забезпечують можливість розробки системи обробки замовлень ;
- Розробити систему замовлень ;

Об'єкт дослідження: процес обробки онлайн замовлень.

Предмет дослідження: засоби створення комп'ютеризованої системи замовлень

Методи дослідження: Методологічною і теоретичною основою дослідження є загальнонауковий аналітичний метод, системний метод, статистичні методи, класифікації, кластеризації даних.

Інформаційну базу дослідження становлять статистичні дані по аналізу логістичної діяльності представлені в глобальній мережі Інтернет.

РОЗДІЛ 1. Системи замовлень

1.1. Основні поняття системи замовлень

Посилання в Wikipedia дає таке значення системи управління замовленнями.

Система управління замовленнями (OMS) - це комп'ютерна програмна система, що використовується в ряді галузей для введення та обробки замовлень.

Але мені більше подобається інше визначення яке дає більш розгорнуту відповідь

Система управління замовленнями (OMS) - це цифровий підхід до управління життєвим циклом замовлення. З її допомогою можна відстежувати всю інформацію і процеси, включаючи надходження замовлення, управління запасами, виконання замовлення і післяпродажне обслуговування. OMS забезпечує прозорість як для бізнесу, так і для покупця. Організації можуть практично в режимі реального часу отримати уявлення про запаси, а замовники можуть відстежувати доставку свого замовлення.

Управління замовленнями - це діяльність, здійснювана в період часу між моментом отримання підприємством замовлення і моментом, коли персонал складу отримує вказівку на відвантаження продукції, щоб виконати замовлення. Також використовується термін «обробка замовлення».

Одне із завдань системи управління замовленнями - це швидко і ефективно обробляти заявки, відстежувати переміщення товарів, здійснювати бухгалтерський та складський облік і оптимізувати ці процеси на будь-якому етапі.

Управління замовленнями в організації починається з прийому та обробки замовлень споживачів матеріальних ресурсів або готової продукції. При цьому використовується інтегрований підхід до управління замовленнями. Він передбачає прийняття оптимальних рішень по закупівлях і запасам матеріальних ресурсів, виробництва, складування і розподілу готової продукції на базі точного прогнозування споживчого попиту.

Це можливо за умови високої готовності організації до поставок замовлених товарів, високої якості сервісу поставок, а також функціонування логістичної інформаційної системи і базових логістичних систем (MRP II, DRP II).

1.2. Основні принципи і переваги впровадження електронних систем замовлення

Перш за все потрібно розуміти що робота над замовленнями завжди лягає на плечі логістичного відділу який має свою логістичну систему і перед тим як заглиблюватися в електронні системи замовлень потрібно розібратись з відділом. Незважаючи на те, що логістика пов'язана в основному з транспортом, потрібно пам'ятати, що логістичним процесам підвласний також матеріальний і інформаційний потік. Проблеми управління розвитком сучасної логістичної діяльності підприємства знайшли своє відображення в чисельних публікаціях вітчизняних та зарубіжних дослідників. Серед вітчизняних науковців відомі праці О. Амоші, Н. Коніщева, Н. Трушкіна. Серед зарубіжних вчених значний інтерес становлять праці С. Кравчика, П. Петраса, Д. Двілінського та інших.

Відділ логістики відноситься до організаційної служби підприємства, до основних напрямків якого слід віднести[5].

- Оптимізація маршрутів переміщення товару. Постачання і доставка товарів має займати якомога менше часу.
- Скорочення витрат. Рациональне використання обігових коштів

- Забезпечення цілісності вантажу. Вданому випадку йдеться про максимальну безпеку вантажу при транспортуванні.
- Забезпечення ефективного зв'язку різних відділів компанії

Обробка замовлень строго пов'язана з інформаційними потоками в логістичній системі і включає ряд операцій. Клієнтам, можливо, доведеться запитувати товари, заповнюючи форму замовлення. Ці замовлення передаються та перевіряються. Потім перевіряється наявність запитуваних предметів та кредитний статус клієнта. Пізніше товари витягуються зі складу (або виготовляються), упаковуються та доставляються разом із їх транспортною документацією. Нарешті, клієнти повинні отримувати інформацію про стан своїх замовлень. Традиційно обробка замовлень займає дуже багато часу (до 70% від загального часу циклу замовлення). Однак за останні роки це значно виграло від прогресу в галузі електроніки та інформаційних технологій. Штрих-код сканування дозволяє роздрібним продавцям швидко ідентифікувати необхідну продукцію та оновлювати записи запасів. Ноутбуки і модеми дозволяти продавцям перевіряти в режимі реального часу, чи є товар на складі, і миттєво вводити замовлення. EDI дозволяє компаніям вводити замовлення на промислові товари безпосередньо в комп'ютері продавця без будь-яких документів.

При розробці логістичної стратегії менеджери прагнуть досягти відповідного компромісу між трьома основними вдосконалення. цілі: зменшення капіталу, зниження витрат та рівень обслуговування Зменшення капіталу. Перша мета - якомога більше знизити рівень інвестиції в логістичну систему (що залежить від обладнання та запасів). Це може бути здійснено різними способами, наприклад, шляхом вибору державних складів замість приватних складів та використання загальних перевізників замість приватних транспортних засобів. Звичайно, зменшення капіталу зазвичай відбувається за рахунок вищих операційних витрат.

Друга мета - мінімізувати загальні витрати, пов'язані з транспортуванням та зберіганням. Наприклад, можна експлуатувати приватні склади та транспортні

засоби (за умови, що обсяг продажу досить великий). Підвищення рівня обслуговування. Рівень логістичного обслуговування значною мірою впливає на задоволеність споживачів, що в свою чергу має значний вплив на доходи. Таким чином, покращення рівня логістичного обслуговування може збільшити доходи, особливо на ринках з однорідною низькою ціною товарів, де конкуренція не заснована на особливостях товару. Рівень логістичного обслуговування часто виражається через час циклу замовлення, визначається як час, що минув від моменту випуску замовлення на придбання (або запиту на послугу) до часу отримання товару замовником (або надання послуги користувачеві). Час циклу замовлення є випадковою величиною з мультиноміальною ймовірністю розподілу. Для ілюстрації, функція щільності ймовірності ланцюга поставок зображена на рис.1.1. Коли торгова точка видає замовлення, можуть статися такі події:

- якщо товари, які вимагає торгова точка, є в асоційованому складі, товар буде доставлений незабаром;
- в іншому випадку склад повинен поповнити свої запаси, розмістивши замовлення в системі виробника, і в цьому випадку відвантаження до роздрібної торгівлі буде затримано;
- якщо товарів немає в наявності навіть у виробника, заводи попросять їх виготовити.

Дозволяє стора, сторb і сторc бути ймовірностями подій a, б і c, і нехай $f_a(t)$, $f_b(t)$, $f_c(t)$ бути (умовною) функцією щільності ймовірності часу циклу замовлення у випадку подій a, б і c відбуваються відповідно. Тоді функція щільності ймовірності часу циклу порядку : $f(t) = p_a f_a(t) + p_b f_b(t) + p_c f_c(t)$.

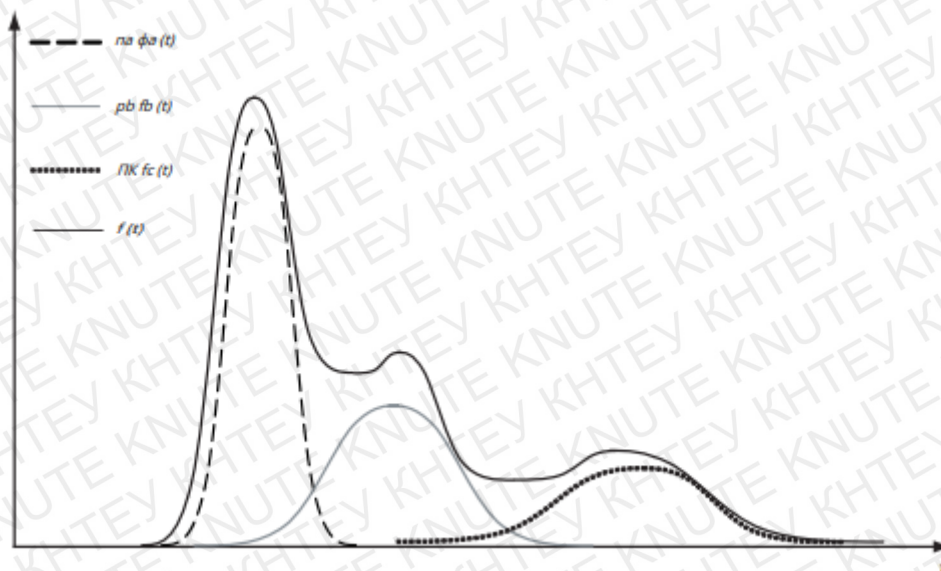


Рис. 1.1 Функція щільності ймовірності часу циклу порядку.

За останні роки кілька стратегічних та технологічних змін помітно вплинули на логістику. Серед них варто згадати три: глобалізація, нові інформаційні технології та електронна комерція.

- Глобалізація. Все більша кількість компаній працює на світовому рівні, щоб скористатися нижчими виробничими витратами або дешевою сировиною, доступною в деяких країнах. Це іноді досягається шляхом придбання або стратегічних союзів з іншими фірмами. В результаті глобалізації зросли потреби у транспорті. Більшу кількість деталей та напівфабрикатів потрібно переміщувати між виробничими майданчиками, а транспортування на ринки, як правило, є більш складним та затратним.
- Інформаційні технології. Постачальники та виробники використовують EDI. Це дозволяє їм обмінюватися даними про рівні запасів, терміни поставок, позиціонування нетранспортних товарів у ланцюзі постачання тощо. На операційному рівні, геоінформаційні системи (ГІС), системи глобального позиціонування (GPS) та бортові комп'ютери дозволяють диспетчерам відстежувати поточне положення автомобілів та спілкуватися з водіями. Такі технології необхідні фірмам, що займаються експрес-

доставкою та доставкою, а також компаніям, що займаються вантажоперевезеннями на великі відстані.

- Електронна комерція. Все більша кількість компаній здійснює комерційні операції через Інтернет. Прийнято розрізняти від бізнесу до бізнесу (B2B) та від бізнесу до споживачів (B2C) транзакції. Зростання електронної комерції паралельно зростанню глобалізації та інформаційних технологій. В результаті електронної комерції обсяг товарів між виробниками та роздрібними продавцями повинен зменшитися, тоді як слід очікувати більш прямих поставок між виробниками та кінцевими споживачами. Електронна комерція призводить до більш складної організації всієї логістичної системи (електронна логістика), які повинні мати можливість управляти малими та середніми відправками великій кількості клієнтів, іноді розкиданих по всьому світу. Крім того, повернення потоку дефектних (або відхилених) товарів стає основною проблемою (зворотна логістика). В системі електронної логістики, як правило, застосовуються різні підходи до функціонування складів та розподілу. віртуальний склад та Точки присутності на території (ROPITТ) - лише кілька прикладів. Віртуальний склад - це приміщення, де постачальники та дистриб'ютори зберігають свої товари на складі таким чином, що компанія електронної комерції може виконувати свої замовлення. ROPITТ - це компанія, що належить компанії, куди клієнти можуть звернутися або для придбання та отримання замовлених товарів, або для повернення дефектної продукції. На відміну від традиційних магазинів, ROPITТ зберігає лише товари, які вже продали товари, які чекають, щоб їх забрали клієнти, та дефектні товари, які чекають повернення виробникам. Це рішення спрощує управління розподілом, але знижує рівень обслуговування споживачів, оскільки не дозволяє здійснювати доставку додому.

Головним завданням управління замовленнями на підприємстві є скорочення часу логістичного циклу замовлень з метою підвищення якості обслуговування

споживачів, зниження рівня товарних запасів і загальних логістичних витрат. Логістичний цикл замовлення в загальному вигляді включає в себе:

- 1) час на формулювання замовлення і його оформлення в установленому порядку.
- 2) час на доставку або передачу замовлення постачальнику.
- 3) час виконання замовлення (час очікування постановки замовлення на виконання, час виконання замовлення, час простоїв, комплексу послуг).
- 4) час доставки виготовленої продукції замовнику.
- 5) час на підготовку продукції до споживання.

Максимальна тривалість циклу виконання замовлень визначається часом, яке споживач готовий чекати з моменту розміщення замовлення до отримання товару (години, дні). Фактичний час виконання замовлення може перевищувати цей час. В такому випадку виникає «розрив часу виконання замовлення». Мета логістичної служби скоротити цей розрив.

Для цього слід або знизити час виконання замовлення у постачальника, або збільшити цикл замовлення споживача за допомогою отримання більш ранній його заявки. Слід зауважити, що етап оптимізації логістичного циклу замовлень є вкрай важливим для діяльності організації, оскільки отримані результати гарантують їй отримання конкурентних переваг.

Одним із методів скорочення часу на обробку замовлень являється впровадження електронних систем обліку, обробки, комунікації і багато чого іншого.

Автоматизоване управління замовленнями починається з такого здавалося б простого кроку, як автоматизація прийому заявок. Немислимо більше вести облікову діяльність вручну на паперових носіях або в імпровізованих формах

таблиць excel. Прибуткове передове підприємство не може дозволити собі подібної халатності.

Як відомо: «час - гроші». Автоматизація системи управління замовленнями клієнтів дозволяють оптимізувати основні процедури продажу. Таким чином, ви отримуєте можливість делегувати весь процес обробки замовлення спеціалізованому ПЗ і заощадити значну частину тимчасових і трудовитрат. Крім того, клієнт тепер може відстежувати статус свого замовлення.

Але не всюди потрібно інтегрувати новітні технології, особливо у сферах де клієнт може вплинути на процес і кінцевий результат свого замовлення. Наприклад товари які створюються на замовлення або товар який має властивості які не передає камера, наприклад меблі або одяг колір і розмір яких важко передати цифровими носіями виведення інформації. І якщо з одягом простіше оскільки є стандарти розмірів по яких можливо орієнтуватись, та й якщо товар раптом не підійшов закон України „Про захист прав споживачів” передбачає повернення товару або його заміна протягом 14 днів з дня укладання договору тобто покупки, то з товарами такими як меблі складніше оскільки даний товар має значні розміри тому на його транспортування потрібно витратити більше коштів. Також часто меблі виготовляються на замовлення враховуючи габарити потрібні для замовника тому даний вид товару підлягає поверненню або заміні лише при виявленні дефектів, в іншому випадку такий товар не підлягає поверненню оскільки такий товар не можна продати іншому клієнту. Продаючи товари виготовлені на замовлення виграють обидві сторони, клієнт отримує унікальний товар виготовлений лише для його а продавець страхує себе від безпідставного повернення товару. Але також є значний недолік оскільки на такі товари не створюють супровідну документацію в якій буде чітко прописані всі норми прийому готового товару. Всі нововведення які впроваджуються в підприємство направлені на оптимізацію і скорочення затрат, але не завжди впровадження електронної системи замовлень призведе до скорочення матеріальних та фізичних

затрат. Перед впровадженням OMS підприємство оптимізує свою CRM систему оскільки саме вона дозволяє допомогти підприємству досягти поставлених цілей і покращує свій фінансовий показник.

Дуже часто деякі люди плутають дві системи: OMS і CRM (Customer Relationship Management). Важливо розуміти, що OMS - це система управління замовленнями, і призначена вона в першу чергу для створення, обробки і управління замовленнями, і саме про це йтиметься. CRM-система ж призначена для управління взаємовідносинами з клієнтами: в ній ведеться облік клієнтської бази, зберігаються особисті дані клієнтів і історія взаємодії з ними, а також міститься різна інформація про завдання, угодах і замовленнях.

Найчастіше, промислові підприємства і постачальники, працюючи з великими федеральними мережами, отримують замовлення через EDI-сервіси (системи електронного обміну даними): замовлення формується в електронному вигляді і надходить в базу даних постачальника. Крім цього, формується ряд різних електронних документів: накладні, акти, рахунки, повідомлення і так далі. Це дуже зручно, швидко і практично.

Але крім роботи з федеральними мережами, постачальники працюють і з дистриб'юторами. І проблема полягає в тому, що велика частина дистриб'юторів до EDI-сервісів не підключена (бо для дистриб'юторів це не вигідно). Тому, найчастіше замовлення подаються через менеджерів за допомогою електронної пошти: або в листі просто повідомляється, на які номенклатури необхідно сформувані замовлення, або заповнюється бланк з номенклатурами і з відповідними обсягами для того чи іншого дистриб'ютора і так само надсилається поштою. Хтось умудряється формувати замовлення по телефону або через месенджери. Це цілком нормальна практика, в цьому немає нічого поганого, але при такій механіці формування замовлень виникає ряд проблем:

- Банально, але замовлення може загубитися в пошті. Таке, хоч і не часто, але буває. Наслідки можуть бути вкрай неприємними, особливо якщо

дистриб'ютор порівняно недавно став співпрацювати з постачальником і ще не може бути повністю впевнений в його надійності: можливо різке зниження замовляються обсягів або навіть повна заміна асортименту на асортимент конкурента.

- Якщо менеджер висилає замовлення в бланку: позиції в бланку можуть бути неактуальними. Заміна однієї позиції на іншу, висновок з асортименту, поява нових позицій, зміна кількості вкладень в одній упаковці тієї або іншої номенклатури: якщо хоч щось у бланку не враховано - бланк вже неактуальне. При завантаженні замовлення в систему, можуть виникнути помилки. Доведеться повідомляти про це менеджера, менеджеру доведеться заповнювати новий бланк, замовлення доведеться приймати заново, а це все - подвійна робота.
- Великі тимчасові витрати на ручне формування замовлення, його обробку і коригування. Потрібно відкрити актуальний бланк, заповнити дані на замовлення, ввести дату доставки клієнту, зайти на пошту і відправити цей бланк. Замовлення необхідно прийняти, перевірити на коректність даних, перевірити актуальність бланка, завантажити замовлення в систему, повідомити менеджера про прийняття і так далі. На це все потрібен час. Якщо замовлення необхідно скорегувати - повторюємо алгоритм, знову витрачаємо дорогоцінний час.
- Неможливість додавання різних модулів типу рекомендованого замовлення або установки примусових обмежень при формуванні замовлень.

У підсумку, для впровадження виходить наступний алгоритм: рекомендується скласти фінансово-економічне обґрунтування, провести оцінку технічних можливостей і коректно розрахувати реальний час впровадження

системи. Також, потрібно написати техзавдання, в якому будуть детально описані цілі, алгоритми, логіка розрахунку того чи іншого параметра і все бажані функції. Потім, необхідно підібрати команду (аутсорсинг або свої програмісти), оцінивши їх кваліфікацію. Після впровадження слід ретельно протестувати систему на наявність помилок. Внести корективи, навчити співробітників (написати інструкції при необхідності) і повноцінно запустити проект.

Система управління замовленнями значно спрощує процес прийому, обробки та коригування замовлень, особливо якщо їх дуже багато. Завдяки їй досягається високий рівень автоматизації і оптимізації: ручна робота зводиться до мінімуму, знижується витрачається час на формування замовлення, а сам процес стає максимально прозорим і зручним.

1.3. Проблеми впровадження електронних систем замовлення

Незважаючи на переваги електронних систем замовлення, існують типові проблеми впровадження OMS.

1. Організаційні складності. Виникають, коли немає чіткого розуміння, навіщо впроваджується OMS, коли персонал не зацікавлений в тому, щоб використовувати нові принципи роботи або не дуже їх розуміє.

При відсутності повного охоплення (половина співробітників підприємства працює з використанням OMS, половина немає) ефективність знижується в рази, тому що затримки і втрати залишаються, а єдина централізована система не вибудовується.

2. Економічні складності. Купівля та налаштування продукту мають помітні витрати, особливо коли фірма потрапляє в "залежність" від розробника і змушена залучати його для вирішення всіх поточних проблем.
3. Технічні. В окремих випадках доводиться додатково витратитися на нове обладнання, сервери, рішення технічних завдань, невіддільних від впровадження OMS.

Окремо варто звернути пильну увагу на вміння і готовність персоналу працювати за новими принципами. Якщо впроваджуваний продукт дуже складний і незвичний, не варто чекати від нього доброго результату, особливо на перших порах.

Управління замовленнями клієнтів за традицією включало в себе ручне введення даних про замовлення і складські запаси. Зрозуміло, помилок при цьому було не уникнути. Загальна ефективність знижувалася ще й через довгого

ланцюжка обслуговування (при русі від відділу до відділу) - а це додаткові витрати, помилки і плюс ризик фальсифікацій.

Не так просто відстежувати виконуються операції в комплексі, коли у всіх відділах різний програмне забезпечення. Без чіткого взаємозв'язку між бізнес-групами процес виходить перевантаженим, занадто витратним. В результаті клієнти залишаються незадоволені (коли, наприклад, оформляють заявку через інтернет-магазин, а вона довгий час залишається невиконаною). Труднощі в процесі управління замовленнями клієнтів можуть виникнути різні.

- Пошук, обробка, реалізація замовлень

Тут важливий вибір місця виконання замовлення (з урахуванням наявної сировини), планування витрат (з урахуванням строків доставки), визначення часу відвантаження, ведення звітності по кожному етапу роботи над замовленням.

- особливі випадки

Маються на увазі порушення термінів доставки. Наприклад, якщо товар раптом активно почали розкуповувати і на складі його не залишилося або замовлення загубився. Тоді обов'язково слід зв'язатися з клієнтом і або скасувати замовлення, або відправити товар заново.

- процедура доставки

Багатьом клієнтам зручно залишати заявку через Інтернет, а потім приїжджати в магазин за своїм товаром. Для самої компанії використовувати магазин як склад - теж великий плюс. Це дозволяє поліпшити оборот, обслужити більше покупців, знизити обсяги уцінок. Хоча логістика при цьому, звичайно, стає складніше.

Управління замовленнями клієнтів здійснюється від моменту надходження заявки і до самого кінця, коли адресат отримує свій товар.

Переглянувши проблеми впровадження електронних систем обробки замовлень можна дійти висновку що більшість проблем пов'язані з нечітким описом самої системи і як з нею працювати в нестандартних ситуаціях.

Проаналізувавши попередній розділ можна дійти висновку що ідеальна комп'ютеризована система замовлень має відповідати таким вимогам:

1. Адаптивність - адаптивність означає здатність програми автоматично пристосовуватися відповідно до різних розмірів екрану і форм-факторів, щоб розумно використовувати доступний простір, забезпечуючи чудовий інтерфейс користувача та загальний інтерфейс на будь-якому пристрої, форм-факторі і розмірі екрану.

2. Інтуїтивний інтерфейс – при роботі з таким інтерфейсом користувач використовує звичні для нього дії особливо не задумуючись над тим що та як зробити, його дії ніби інтуїтивні, звідки й назва.

3. Низька вимога до системних ресурсів – це означає що програма не потребує високих апаратних і програмних показників пристрою.

4. Надійність - здатність системи підтримувати визначену працездатність у заданих умовах.

Чим більше цих пунктів буде дотримано тим більш конкурентноспроможною буде система.

РОЗДІЛ 2. Організація розробки комп'ютеризованої системи замовлень

2.1 Опис проблем у сфері обслуговування замовлень

Розглянемо основні етапи управління замовленнями на прикладі виробничого підприємства.

При використанні системи замовлень введення замовлення в базу даних здійснюється значно простіше оскільки користувачу який збирається вносити замовлення в базу даних лише потрібно заносити інформацію про замовлення в чітко визначені поля на стандартним алгоритмом. На цьому етапі вже формується загальне уявлення про ринковий попит, про готовність компанії до його задоволенню, а також якийсь споживчий портрет.

Далі на замовлення плануються управлінські транзакції (отримання і розподіл грошових коштів на роботи на замовлення).

На підставі портфеля замовлень планується короткостроковий, а у випадку з тривалим замовленням (наприклад, НДДКР) довгостроковий бюджет.

Після затвердження замовлення він ділиться на частини, по кожній з яких формується конкретне завдання. Якщо замовлення не має на меті виготовлення товару а лише продаж вже готового товару користувачеві ділення замовлення на частини непотрібно.

За цим завданням формуються внутрішні замовлення, які здійснюються між структурними підрозділами підприємства. Наприклад, замовленням відділу продажів виробничому може бути постановка в виробництво певної кількості найменувань товарно-матеріальних цінностей. Такі замовлення не оплачуються підприємством, так як вони створюються і виконуються всередині одного

підприємства. Найчастіше вони служать для забезпечення підрозділів ТМЦ, необхідними для їх безперебійного функціонування.

Звичайно, для виконання певних етапів замовлення потрібно рух грошових коштів, але це більше пов'язано із зовнішнім забезпеченням (розрахунки з субпідрядниками, постачальниками і т.д.).

Якщо перед менеджером постає виконання декількох замовлень то менеджер орієнтується на більш значущі, наприклад по термінам або об'єму.

Після закінчення робіт над замовленням менеджер готує замовлення і супроводжуючі папери на доставку товару до клієнта і його отримання.

При веденні управлінського та бухгалтерського обліків основними є фінансові потоки і їх відображення у вигляді реалізації і відпустки ТМЦ зі складу, рух грошових коштів, нарахувань премій за роботу, формуванні дебіторської і кредиторської заборгованості. Після того як всі дані прийняті до обліку та супроводжуюча документація готова, менеджер зв'язується з клієнтом і виробляє відвантаження.

Тобто менеджер виконує контролюючу роль, проводить контроль за виконанням замовлення від початку прийняття до завершення. Його обов'язком є контроль над ситуацією із замовленнями: відправка замовлення в виконання, визначення причин простою, оповіщення клієнта про готовність. Він несе відповідальність за те, щоб всі структури злагоджено відпрацювали і на виході вийшов готове замовлення.

Дії менеджера регламентовані, а переміщення замовлення описані бізнес-процесами підприємства. Проте, завжди є ризик, що щось може піти не так. На цей випадок можна запросити консалтингову компанію, яка допоможе з оптимізацією бізнес-процесів і при необхідності перерозподілить терміни або дасть більш детальний опис деяких процесів, а деякі зовсім прибере.

Головна мета впровадження програми управління замовленнями клієнтів - скорочення термінів виконання і витрат, збільшення ефективності роботи менеджерів, короткострокове і середньострокове планування бюджету.

Ефективна робота менеджера оцінюється виходячи з його здатності координувати роботу всіх структурних одиниць бізнес-процесу (замовлення): забезпечити необхідний контроль роботи внутрішніх підрозділів (виробничих і фінансових відділів) і зовнішніх виконавців. Від цього безпосередньо залежить ефективність компанії в цілому.

Через стрімку діджиталізацію правила на ринку надання послуг суттєво змінилися порівняно з двухтисячними роками. Конкуренція у всіх сферах ринку суттєво зросла оскільки на ринок вийшли фізичні підприємці які орієнтуються на певних клієнтів. Поглянувши на рекламу або інтернет магазини в соцмережах можна побачити що попит на товари від невідомих виробників значно більший ніж попит на аналогічний товар у відомого виробника оскільки на даний момент клієнт цінує індивідуальний підхід і неповторність яку не може дати великий бренд, та й ціна від невідомого виробника в декілька разів дешевша. Але перед фізичними підприємцями постає проблема, як працювати з клієнтом яких знаходиться в іншому місті, країні або навіть континенті. З цього витікає безліч проблем для вирішення яких потрібно затрачувати матеріальні та трудові ресурси яких у фізичної особи підприємця не так багато. В залежності від того в якій сфері підприємець вирішив відкрити свій бізнес він стикнеться з специфічними проблемами які притаманні певній сфері торгівлі яких до слова налічується чотири види:

- Гуртова торгівля
- Роздрібна торгівля
- Внутрішня торгівля
- Міжнародна торгівля

Торкнемся кожного виду. Гуртова торгівля, або її також називають оптовою торгівлею. Гуртовою торгівлею в основному займаються великі торговельні фірми, торгові дома, дистриб'юторські фірми тощо, які мають обмежений асортимент товарів але в великих об'ємах, не рідко гуртовою торгівлею займаються підприємства які і виготовляють продукцію. За особливостями організації торговельного процесу розрізняють дві основні форми гуртової торгівлі: гуртова торгівля, здійснювана торговельними підрозділами товаровиробників у формі прямого збуту, та гуртова торгівля гуртовими торговельно-посередницькими підприємствами (рис.2. 1). При цьому серед останніх є фірми, які купують товар, отримують право власності на нього та здійснюють перепродаж. Інші гуртові посередники не набувають права власності на товар, а лише виступають організаторами товарного обігу. Велике різноманіття видів гуртової торгівлі дає змогу товаровиробникам і споживачам робити свідомий вибір найефективнішого й зручного каналу організації продажу товарів.

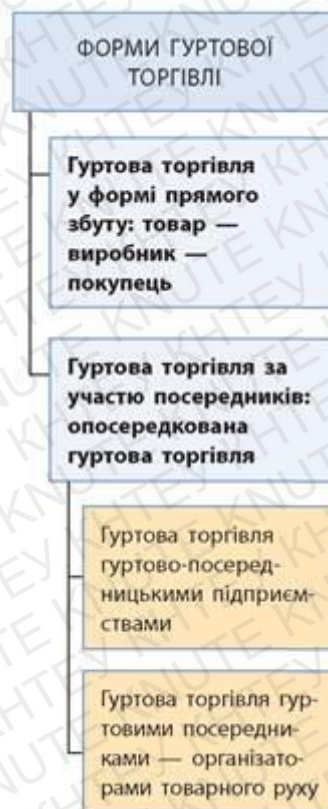


Рис. 2.1. Форми гуртової торгівлі.

Рекламу замовлену цими продавцями ви не побачите на білбордах оскільки компанії які займаються гуртовою торгівлею орієнтуються на специфічну аудиторію яка не потребує рекламної агітації оскільки для її зацікавлення потрібно мати лише достатню кількість товару належної якості з конкурентною ціною. Процес отримання і оплата товару проходить наступним чином: клієнт сам має забрати замовлений товар на складі, після отримання товару клієнт має розрахуватись за товар на місці, найчастіше розраховуються готівкою, або за згодою продавця може проходити онлайн оплата. Також продавець може запропонувати транспорт для доставки за певну орендну плату.

Підприємства які займаються гуртовою торгівлею використовують перевірені ресурси для управління компанією за які готові платити. Такі компанії майже не мають проблем в сфері обробки замовлень оскільки єдине що від них потрібно це наявність потрібного товару на складі в достатній кількості та якості.

Клієнтами гуртових продавців являються роздрібні торговці які купують партію товару в гуртових торговців і продають роздрібно в точках збуту. В роздрібній торгівлі клієнтами являються кінцеві споживачі продукту. Роздрібна торгівля — це продаж товарів безпосередньо споживачам у магазинах, на ринках, підприємствах харчування. Особливість роздрібною торгівлі полягає в тому, що тут товари реалізуються в невеликій кількості. Із часом роздрібна торгівля почала зосереджуватися у великих торговельних центрах, які називають супермаркетами, гіпермаркетами, торговельними молами, аутлетами та ін. Роздрібні торговці зачасти це фізичні особи підприємці які мають свою точку збуту в вигляді магазину. Також з 2020 року набула популярності незареєстрована підприємницька діяльність яка проводиться через соціальні мережі. Найбільшої популярності даний вид торгівлі набув за час карантину, після чисельних скорочень рівень безробіття зріс з 8,5% до 9,9%, а також відбулось падіння реальних доходів українців. На фоні цих подій громадяни почали шукати додатковий дохід на який не вплине погіршення карантинних обмежень[1].

Оскільки незареєстрована підприємницька діяльність незаконна, розглянемо лише тих хто займається цим легально й офіційно. Тож порівняно з гуртовими торговцями для максимального отримання замовлень роздрібні торговці не рідко використовують методи маркетингу для приваблення клієнтів. Для підвищення конкурентоспроможності роздрібні торговці повинні постійно вводити нові види актуальних товарів, знижувати ціни на продукцію або послуги, широко використовувати рекламні заходи, реалізувати товар через більшу кількість торгових точок та впроваджувати інновації. Основною перевагою в 2021 році стоїть швидкість обробки замовлень оскільки клієнтами являються споживачі даного товару які купують товар по мірі необхідності і для них строк з моменту замовлення до отримання товару має велике значення. Головним гаслом для таких клієнтів являється вираз «Хочу тут і зараз!» за який вони готові переплатити. Наглядний приклад тому це магазини які знаходяться безпосередньо біля житлових комплексів і магазини які знаходяться далі від житлових будинків. В магазинах які знаходяться безпосередньо біля житлових комплексів ціни на продовольчі товари будуть дорожчі ніж в магазинах які знаходяться далі від житлових будинків хоча попит в перших буде більший оскільки клієнти готові переплатити за товар якщо він знаходиться ближче.

Тому всі роздрібні торговці орієнтуються на внутрішній ринок оскільки при міжнародній торгівлі є багато нюансів про які ми поговоримо пізніше. При внутрішній торгівлі збут товару проводиться виключно в межах країни, тобто товар не проходить митницю при доставці. При міжнародній торгівлі доставка товару може затягуватись до декількох місяців не кажучи вже про велику вартість доставки.

2.2 Мета і призначення комп'ютеризованої системи замовлень.

Метою даної розробки є побудова комп'ютеризованої системи замовлень.

Підставою для розробки системи замовлень є завдання на дипломне проектування, затверджене кафедрою комп'ютерних наук та інформаційних систем Національного технічного університету України «Київський національний торговельно-економічний університет»

Створити універсальну систему замовлень неможливо оскільки для цього потрібно буде засунути в програму безліч функцій які ускладнять навігацію по інтерфейсу та збільшать мінімальні системні вимоги тому перед розробкою системи потрібно чітко розуміти для кого її розробляти і в яких випадках її будуть використовувати. Перш за все потрібно проаналізувати доцільність створення такої системи оскільки на ринку може вже існувати така система й купити готову систему буде доцільніше ніж її створити. Для цього переглянемо самі популярні пропозиції серед систем які пропонують різні компанії.

Переглянувши багато рейтингів CRM систем на різних сайтах та форумах я склав свій власний рейтинг систем які задовольняють більшість вимог які потрібні середньостатистичному користувачу даної системи

• Бітрікс24

CRM для середнього бізнесу і великих проектів створена спільними силами Бітрікс і 1С. Це професійна система з повним набором інструментів для організації спільної роботи компанії: контакт-центр, завдання та проекти, продажу, соціальна мережа фірми, конструктор сайтів. Існує як хмарна, так і версія коробочки.

На російському і українському ринках Bitrix24 впевнено входить в трійку кращих рішень для бізнесу завдяки великим функціональним можливостям (їх кількість безпосередньо залежить від обраного тарифного плану) і простоті управління. З недоліків варто відзначити закритість кастомізації. Наприклад, в

безкоштовної версії можна додати лише одного адміністратора, а при підключенні найбільш дорогого тарифу кількість адміністраторів необмежено.

Основні можливості:

- автоматична звітність;
- онлайн-конструктор документів;
- генератор продажів - може автоматично пропонувати існуючим клієнтам (як усім, так і окремим групам) певні акції;
- маркетингові функції - email-розсилки, sms, персоналізовані рекламні пропозиції для посилення первинних і повторних угод;
- зручний внутрішній чат і відеодзвінки;
- інтеграція з телефонією, календарями, соціальними мережами, 1С;
- функціональне мобільний додаток, сумісний з усіма пристроями.

• **Pipedrive**

Хмарна CRM для малого бізнесу, створена досвідченими менеджерами з продажу в 2010 році. Зараз система повноцінно працює в 140 країнах світу. Функціонал спрямований переважно на управління продажами, маркетинг, спільну роботу і звітність. Однак в функціоналі не вистачає можливості інтеграції з бухгалтерією, тому документальна підтримка продажів відсутня. При роботі з клієнтами виставляти рахунки і вести контракти буде важко.

Система не підійде великим компаніям через відсутність повного циклу продажів від збору першої інформації про клієнта до післяпродажного обслуговування. Однак для малого і середнього бізнесу це непоганий варіант з прийнятною вартістю і хорошим рівнем безпеки даних користувачів.

Основні можливості:

- легка настройка і швидке впровадження;
- більше 150 додатків і інтегрованих модулів для налаштування програми під свої потреби;
- фокусування на завершення угод;
- синхронізація з поштовими сервісами;
- детальна і зрозуміла звітність для аналітики результатів і процесу продажів;
- простий і зрозумілий інтерфейс;
- високий рівень безпеки даних та регулярне створення резервних копій.

• amoCRM

Зручна CRM для малого і середнього бізнесу з простим інтерфейсом для ефективної взаємодії з клієнтом на всіх етапах співпраці. Систематизує входячі заявки, створює персональні рекламні пропозиції і навіть автоматично формує воронку продажів.

Функціонал дозволяє ставити завдання співробітникам, складати докладні плани справ і спілкуватися в чаті, а навчання роботі з нею можна провести за пару годин.

Основні можливості:

- автоматична фіксація заявок з усіх джерел - від форми на сайті до дзвінка і повідомлення в месенджері;
- інтеграція з сервісами розсилок, Лендінзі, соціальними мережами, телефонією, Google Analytics;
- відкритий API;
- мобільний додаток, адаптоване для повноцінної роботи зі смартфона;
- надійність і безпеку з можливістю обмеження прав доступу різним користувачам.

• 1С:CRM

CRM-рішення на платформі "1С: Підприємство" випускаються більше 10 років. Більше 50000 користувачів використовують рішення "CRM" у своїй щоденній роботі, серед них понад 5000 користувачів в Україні. Базова система для управління бухгалтерією яку використовують в 90% підприємств від мікропідприємств, чисельністю менше десяти працівників, до великих компаній з чисельністю працівників більше двохсот п'ятдесяти.

Весь багаторічний досвід впроваджень на підприємствах різних галузей, масштабів і форм власності дозволив перетворити програмний продукт "CRM" в зручний, функціональний, гнучкий інструмент управління взаємовідносинами з клієнтами.

Рішення "CRM" (Customer Relationship Management - Управління відносинами з клієнтами) дозволяють організувати ефективну роботу відділів продажів, сервісного обслуговування, логістики, маркетингу, аналітичної служби, служби якості та інших підрозділів на всіх етапах взаємодії з клієнтами і всередині компанії.

Основні можливості:

- Доброзичливий і спрощений користувальницький інтерфейс , що дозволяє налаштовувати робочі місця користувачів відповідно до їхніх прав у вирішенні, а також загальних налаштувань програми.
- Управління клієнтською базою, докладна характеристика про кожного клієнта і контактну особу, динаміка зміни стану відносин з клієнтами, можливість швидкого введення і доступу до інформації про клієнта;
- Управління контактами з клієнтами, облік історії контактів з клієнтами, реєстрація потреби клієнтів, оперативна передача інформації між відділами, планування контактів;

- Сервісне і гарантійне обслуговування . Облік товарів, що знаходяться на обслуговуванні: по серійних номерах, строкам і типам обслуговування, управління сервісними зверненнями клієнтів; управління сервісними замовленнями зверненнями.
- Захист інформації , настройка для користувачів прав доступу до інформації, забезпечення доступу користувачів тільки до інформації по своїм клієнтам;
- Полегшення виконання рутинних операцій , інтеграція з електронною поштою, підготовка звітів, помічник введення нових клієнтів, пошук двійників клієнтів, групова обробка клієнтів, фільтри.
- Можливість отримання сповіщень через: нагадування в рішенні, e-mail, SMS, індивідуальна настройка повідомлень планованих / прострочених подій в календарі, нових / прострочених завданнях (доручення, інші бізнес-процеси), створення індивідуальних нагадувань (скарги, події і т.д.). Зручна настройка сповіщень користувачем індивідуально.
- Розділ інтерфейсу "Методики" , що містить рекомендації щодо вирішення різних завдань в програмі.

Якщо прискіпливо поглянути на цей рейтинг то можна побачити що всі ці системи потребують певної підготовки підприємства.

- Для початку потрібно підготувати апаратну складову яка буде досить багато коштувати особливо якщо це велике підприємство з великим оборотом коштів.
- Потрібно навчити персонал користуватись системою. Навчання персоналу займе певний час та потягне фінансові витрати. Для великих компаній ці витрати незначні та оправдані особливо якщо ця компанія вже закріпилася на ринку. Але якщо компанія тільки вийшла на ринок і не має достатніх коштів.

- Потрібно оплачувати заробітну плату кваліфікованим працівникам яка буде більшою порівняно з працівниками менш кваліфікованого рівня.

Також більшість систем належать до хмарних сервісів, робота з ними проходить через браузер тому слід подбати також про інтернет з'єднання а це ще додаткові затрати. В великому підприємстві витрати на встановлення системи відіб'ються за декілька місяців але для мікропідприємства витрати на систему можуть викликати значні фінансові проблеми, та й не всім підприємствам потрібна така система, для багатьох підприємств все що потрібно від системи це максимальна простота, максимально низькі технічні параметри та мінімально потрібний набір інструментів для роботи з замовленнями, так працівник не зможе нашкодити системі в міру своєї необізнаності.

РОЗДІЛ 3. Практична реалізація комп'ютеризованої системи замовлення

3.1. Розробка користувацького інтерфейсу

Я вирішив розробляти систему для підприємства яке має свою точку збуту і проводить реалізацію товару лише на своїй точці, при цьому значна частина асортименту знаходиться на складі і представлений в точці збуту лише в вигляді примірника або в каталозі, тому купити товари в точці збуту неможливо, лише замовити. Наприклад "Євродім" які спеціалізуються на продажі корпусних меблів. Продаж всього товару відбувається в точці збуту за адресою: Біла Церква, вул. Ярмаркова 43/24, індекс 09107, при цьому доставка товару відбувається зі складу, а сама точка збуту має лише експонати які представлені для візуальної презентації товару потенційному покупцеві. Систему я вирішив розробляти для складських працівників які повинні отримують готове до виконання замовлення і готують його на відправку. Сама система повинна мати мінімалістичний та інтуїтивний інтерфейс а також знаходитись на комп'ютері працівника і також не має залежати від соціальної мережі. Оскільки в Україні більшість складів знаходяться за межами або на околицях населених пунктів тому не завжди є змога провести міжнародну комп'ютерну мережу. Система повинна займати мало місця на диску. В разі необхідності система повинна в короткі терміни бути адаптована як під базу даних Microsoft Access так і під Microsoft SQL Server. Оскільки це найчастіші сервіси створення замовлень.

Середовище програмування я вибрав Embarcadero RAD Studio. І мову програмування Object Pascal. Embarcadero RAD Studio - це найшвидший спосіб створювати і оновлювати інтенсивно працюючи з даними, розподілені і сильно взаємодіючі додатки з розвиненим і візуально привабливим для користувача інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за

допомогою Object Pascal і C ++. В ній можна з легкістю оновлювати VCL- і FMX-додатки для Windows 10, використовуючи нові елементи управління і VCL-стилі для Windows, а також компоненти служб універсальної платформи Windows. Я вибрав це середовище оскільки вважаю що це найкраще рішення при роботі з базами даних і сервісами Microsoft Office. А мова програмування Object Pascal, це найкраща мова для створення баз даних оскільки мова підтримується багатьма компаніями і сторонніми виробниками ПЗ тому мова знаходиться в постійному розвитку хоча й не так популярна як інші мови.

Перед створенням системи я переглянув інтерфейс інших систем і дійшов висновку що вони універсальні, тобто вони мають всі інструменти які тільки можуть хоч колись знадобитися, але не всі ці інструменти потрібні користувачу, більшість з них лише заважають орієнтуватися в системі. Такі системи створюються для того щоб задовольнити максимальну кількість покупців цієї системи оскільки вони створені для отримання прибутку за допомогою продажу цієї системи. Але оскільки я збираюся розробляти систему під певні потреби потрібно чітко їх описати і потім під ці потреби підібрати інструменти інтерфейсу.

Тож оскільки система розробляється для складської працівників робота яких полягатиме лише в виконанні замовлень які вони будуть брати з бази даних то потрібно поглянути яка їх роль в виконанні замовлення (рис. 3.1).

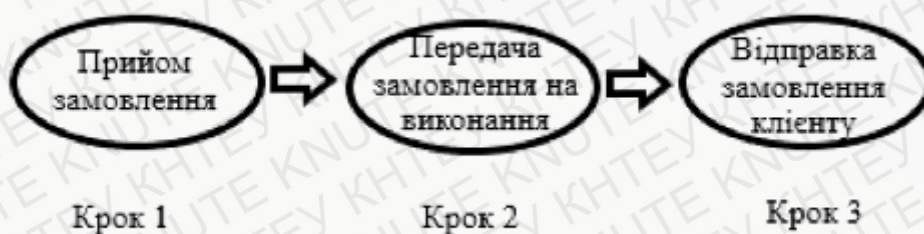


Рис. 3.1 Етапи обробки замовлення.

Перший та другий крок виконують спеціалісти які мають досвід роботи з системами та комп'ютерною технікою взагалі, тому для них потрібна система яка має безліч можливостей та потенціалу. В основному два перших кроки виконують офіс-менеджери які і приймають замовлення від клієнта, формують його та віддають на виконання. Зазвичай для цих кроків використовують. 1С:Підприємство. Але третій крок зазвичай виконують складські робітники, наприклад комірник або завідувач складом. Ці робітники зазвичай в більшості займаються фізичною працею і заробляють менше ніж працівники які виконують перший і другий крок тому і досвід роботи з комп'ютерними приладами в них недостатній для користування сучасними системами обробки замовлень, та й в більшості переглянутих систем відсутній розподіл по ролям і доступу до даних тому більший ризик витоку або пошкодження даних.

Враховавши всі можливі ризики я описав ідеальні параметри інтерфейсу системи а саме:

1. Всі дії зміни в базі даних мають виконуватись лише після натискання кнопок які мають бути підписані літерами.
2. Інтерактивний інтерфейс має бути зведений до мінімуму.
3. Кольори інтерфейсу мають бути м'які, тобто блідих відтінків або невиразних кольорів.
4. Інтерфейс має бути статичний та не піддаватись редагуванню користувачем.
5. Кнопки які виконують однакові дії на різних вкладках повинні виглядати однаково та розташовуватись безпосередньо в полі зору користувача.
6. Інтерфейс який користувач буде використовувати нечасто повинен розташовуватись якомога далі від робочої зони щоб уникнути випадкових натискань.

Для розуміння які саме елементи інтерфейсу потрібно створити для системи замовлень потрібно проаналізувати які дії користувачу потрібно виконувати для обробки замовлень. Оскільки основні обов'язки складських працівників це підготовка замовлення на відправку то для них потрібні лише базові елементи такі

як додавання, видалення та редагування замовлення, також так як в місці де розташований склад товар може бути відсутнє постійне з'єднання з інтернетом то база даних з замовленнями може передаватися на склад фізичними носіями інформації або відсилатися файлом через корпоративну пошту тому розташування файлу з базою даних може знаходитись в різних місцях. Щоб уникнути помилки синхронізації системи з базою даних потрібно дати можливість користувачу вибирати розташування бази даних але щоб уникнути проблеми у вигляді того що користувач вибере файл невідповідного формату, наприклад (.mp3, .jpg,) тому потрібно щоб при виборі файлу користувачу відображалися лише файли формату (.acddb).

Перед додаванням нового елемента інтерфейсу поглянемо на візуальну схему роботи комірника (рис. 3.2).



Рис.3.2. Етапи обробки замовлення складським працівником

На першому етапі працівник отримує замовлення не виконання яке знаходиться в базі даних. Він роздруковує бланк відпуску товару і або список потрібного на відправку товару. Далі він йде на склад і шукає потрібний товар, після того як він знайшов потрібний товар він завантажує його в автомобіль щоб в подальшому доставити клієнту. Для ведення бухгалтерії і щоб у разі недостачі чи претензій збоку клієнта що товар не був доставлений потрібно щоб клієнт особисто розписався на бланку отримання товару. Тому в систему потрібно додати можливість друкувати різні бланки які вже мають мати чітку структуру.

Під час розробки інтерфейсу я брав приклад з базових програм роботи з інформацією, їх інтерфейс доволі одноманітний тому при переході з однієї бази на іншу більшість користувачів інтуїтивно розуміють як з нею працювати (рис. 3.3, рис. 3.4, рис. 3.5, рис. 3.6).

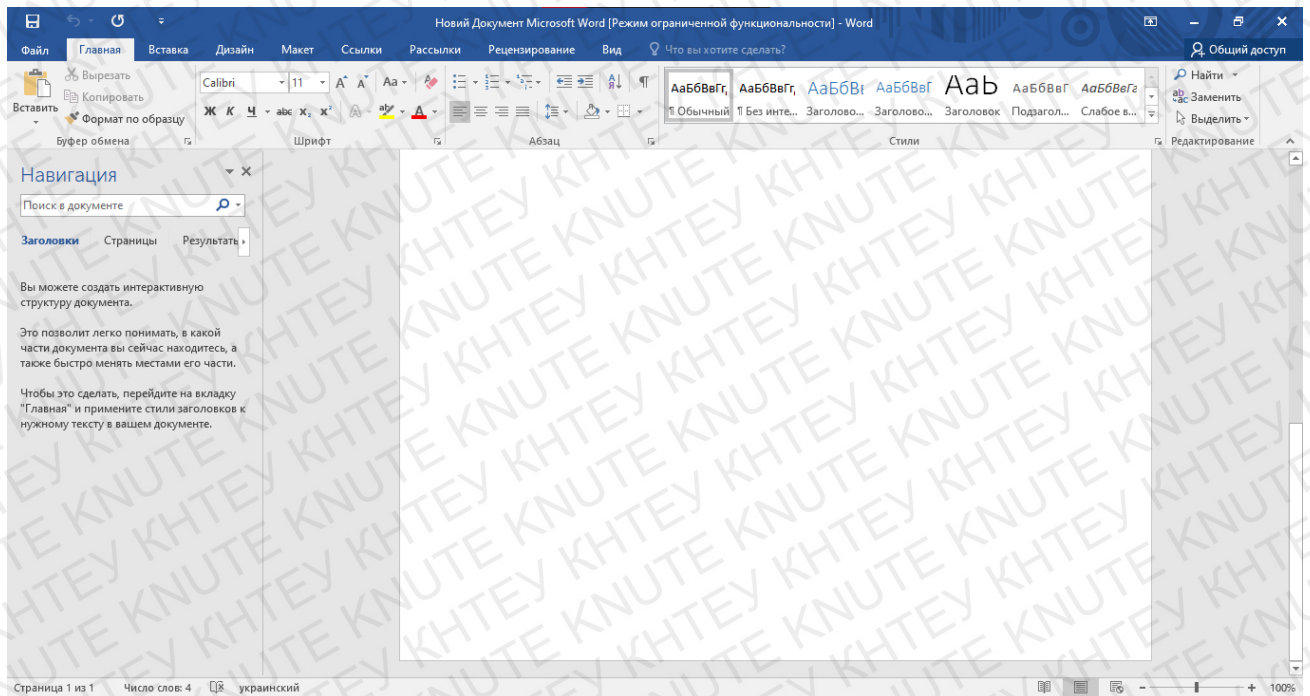


Рис.3.3 Microsoft Word

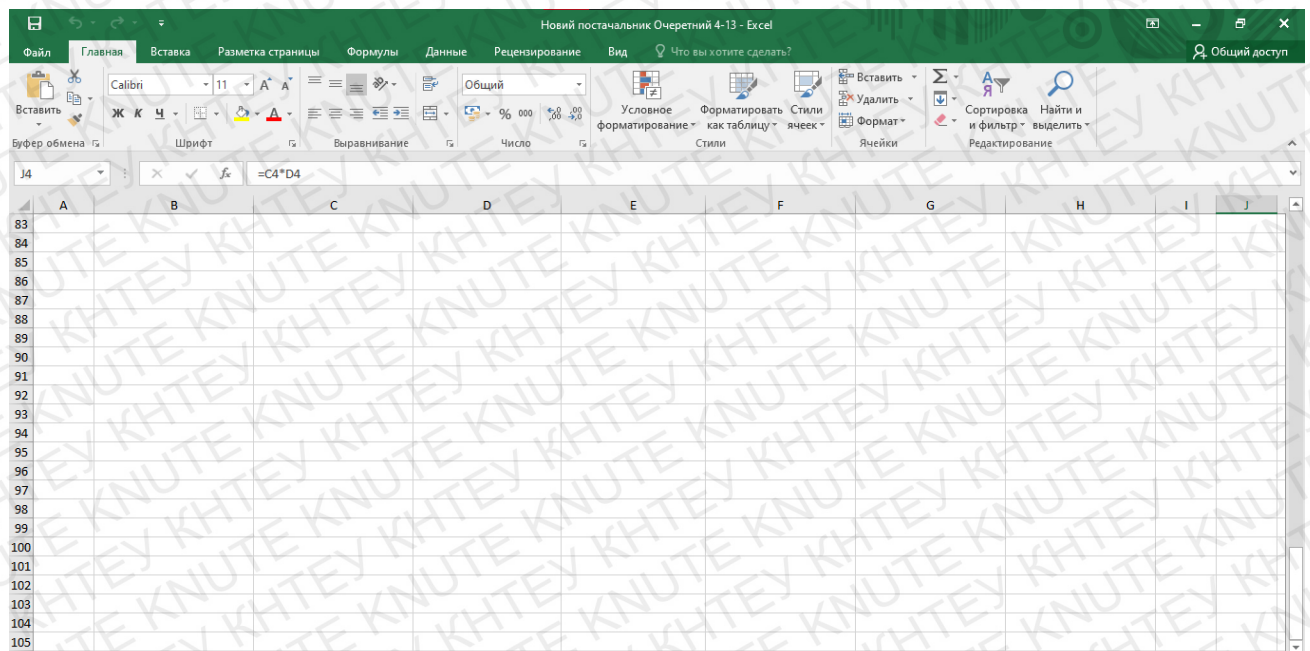


Рис.3.4 Microsoft Excel

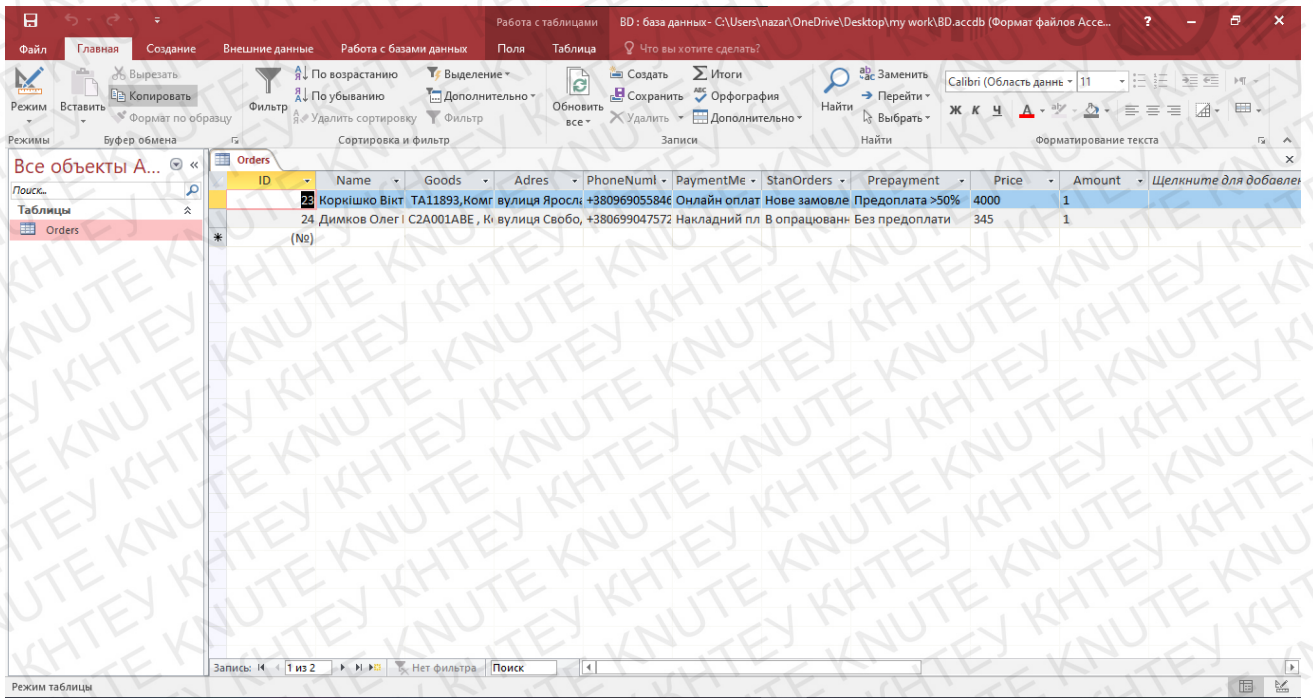


Рис.3.5 Microsoft Access

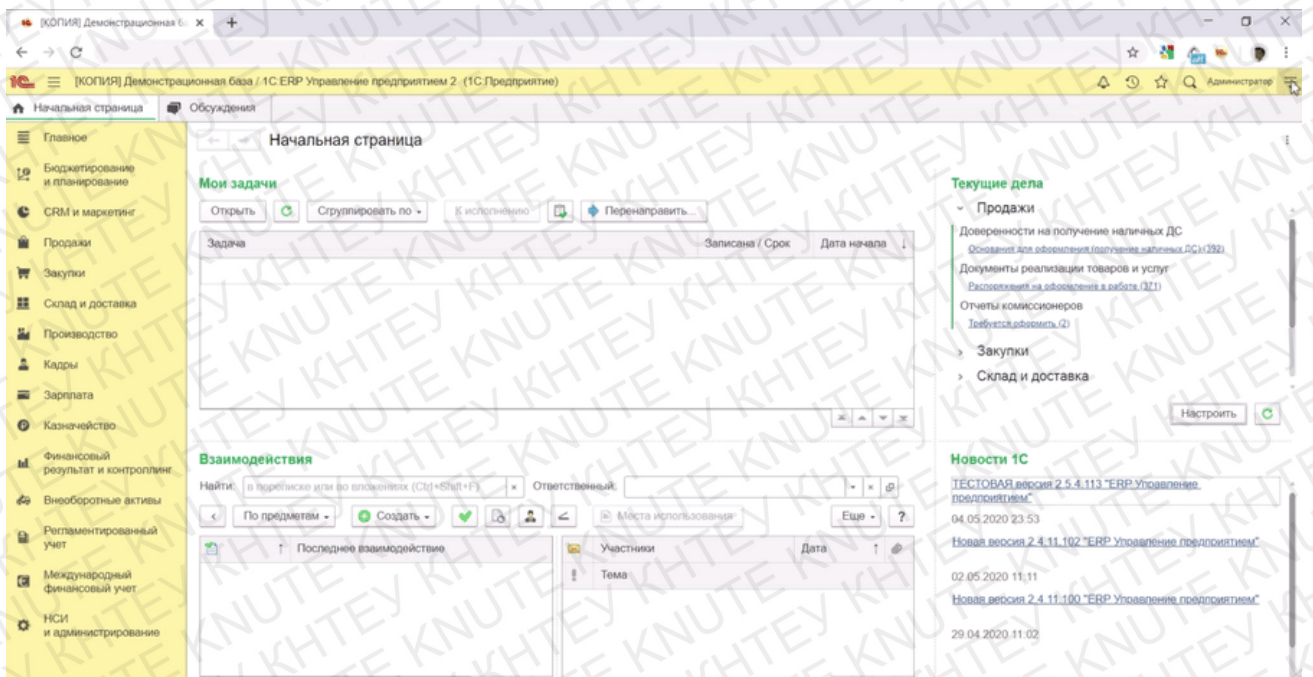


Рис. 3.6 1С: Підприємство

Надихнувшись цими програмами я вирішив створити подібний інтерфейс і ось що з цього вийшло (рис. 3.7).

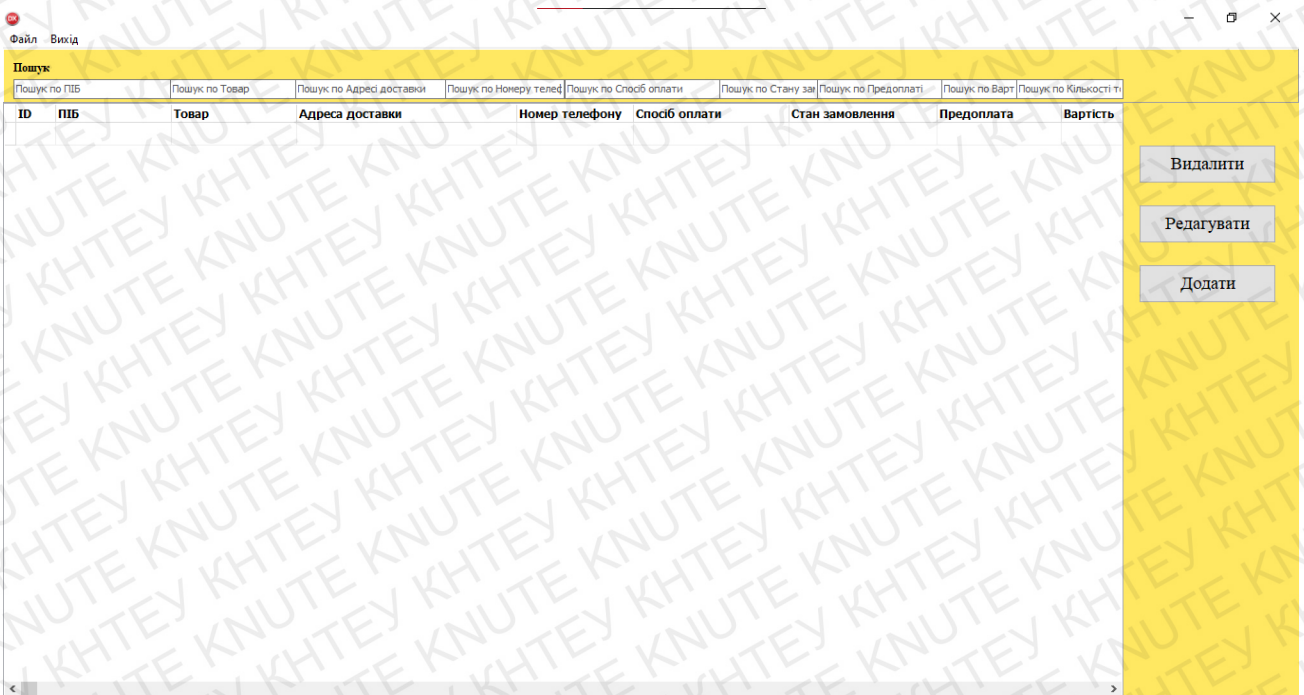


Рис. 3.7 Інтерфейс моєї системи замовлень

Дане середовище лояльніше ніж інші відноиться до користувача і допомагає йому, воно має набір інструментів які можуть знадобитися користувачеві наприклад кнопки поля таблиці і багато інших елементів тому користувачу непотрібно прописувати кожен елемент, за його це зробить середовище, йому лише залишається прописати команди і операції з елементами, наприклад поглянемо як створити програму з нуля, для цього потрібно відкрити середовище програмування, воно відкривається як і люба стаціонарна програма. Після відкриття система відразу відкриває вкладку на якій дає змогу вибрати між створенням нової програми або запуском вже створеної якщо така є, також якщо ви збираєтесь створити нову програму з нуля то на такий випадок програма дає багатий вибір для створення програм різного призначення (рис. 3.8).

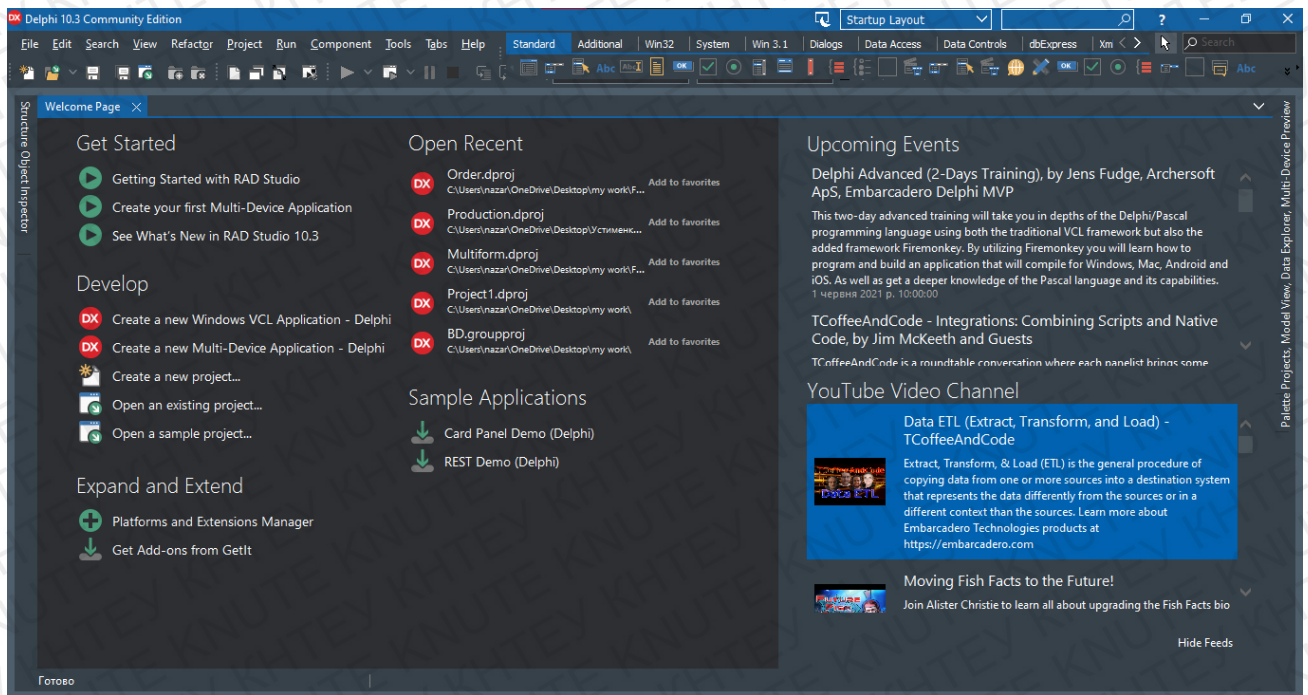


Рис. 3.8 Інтерфейс RAD Studio

Оскільки потрібно створити систему то в цьому вікні потрібно вибрати напис “create a new project”. Після цього відкриється вікно яке дає змогу вибрати призначення майбутньої програми. RAD Studio дозволяє створювати велетенську кількість програм різноманітного призначення починаючи від калькулятора на андроїд закінчуючи іграми з 3D-графікою. Тому на цьому етапі потрібно чітко розуміти який вид вибрати для майбутньої програми, для моєї програми я вибрав звичайну програму для Windows (рис. 3.9).

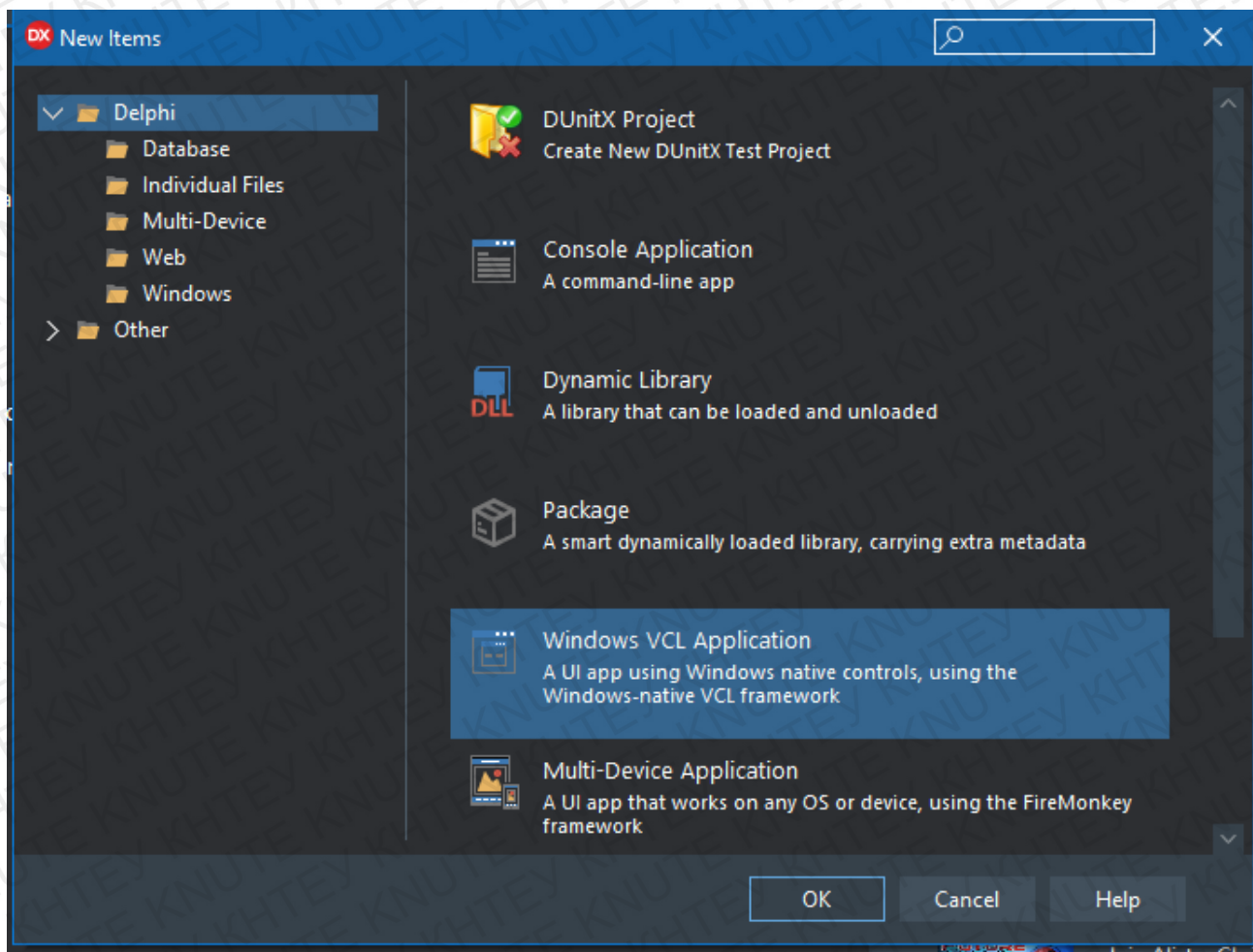


Рис. 3.9 Вибір потрібного виду програми

Після цього відкриється вікно з формою на якій вже можна працювати. Форма це головне вікно програми, з цього боку RAD Studio трохи схожий на звичайний сайт, а саме аналогію можна провести з вебсторінками, в RAD Studio роль сторінок замінюють форми. Весь інтерфейс середовища програмування складається з елементів якими користувач користується протягом більшості свого часу (рис. 3.10).

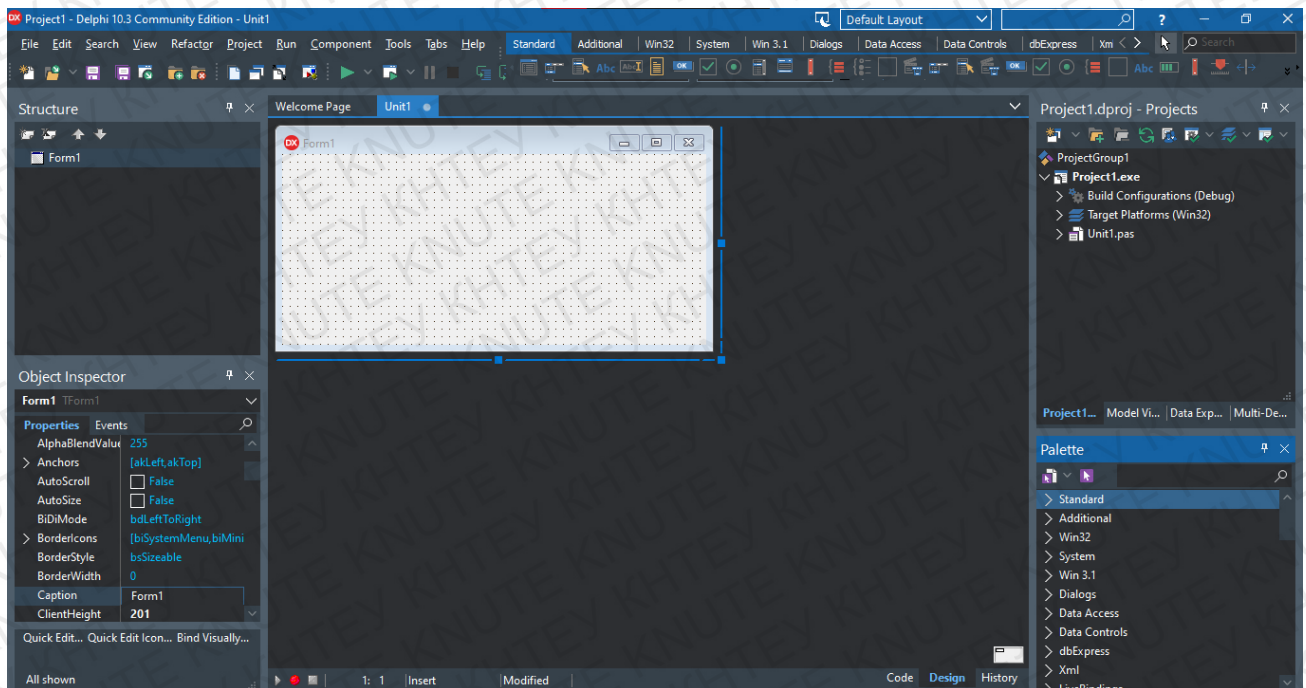


Рис. 3.10 Робоче вікно RAD Studio

Посередині знаходиться сама форма, справа і зверху палітри в яких знаходяться компоненти які можна додати на форму, вони посортовані по категоріям для кращого орієнтування. Ліворуч знаходяться властивості вибраного компоненту, наприклад назва, колір, форма, розмір, текст який підписує даний елемент на формі, та безліч властивостей які можна налаштувати. Якщо двічі клацнути на елемент лівою кнопкою миші, в тому числі і на форму, то відкриється код і для елемента по якому клацнули створиться подія “OnClick”, це подія яка виконається в разі натискання по формі. Взагалі подій безліч, їх можна переглянути на вкладці “Events” яка знаходиться ліворуч від форми (рис. 3.11).

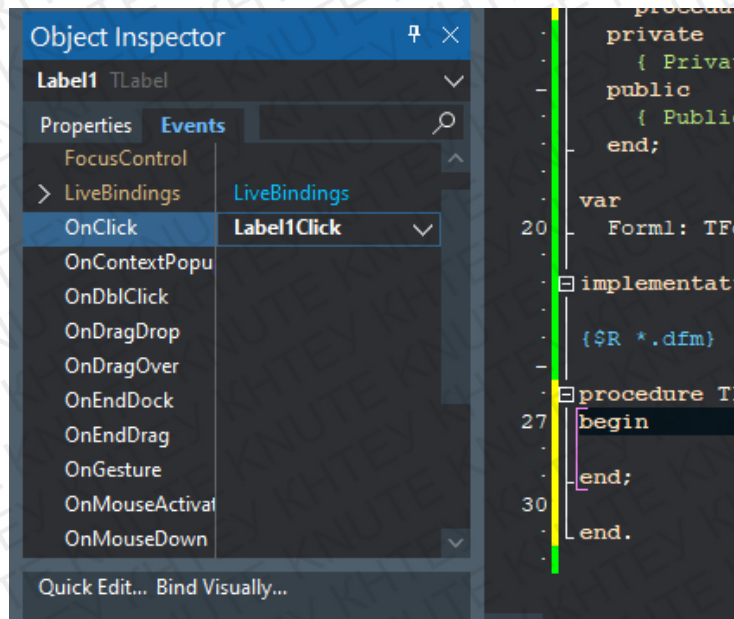


Рис. 3.11 Вкладка “Events”

Кожна ця подія може бути описана, для цього користувач може прописати в коді все вручну або двічі клацнути мишею навпроти потрібної події і середовище відразу пропише цю подію в коді, потрібно буде лише описати що саме повинно виконуватись в разі дій користувача якщо ці дії запускають подію.

Звичайно, у кожного компонента свій набір подій, але майже у кожного візуального компонента події однакові:

- onClicк- виникає при натисканні мишкою на компоненті;
- onExit - виникає, коли компонент втрачає фокус введення;
- onDbлClicк - виникає при подвійному натисканні мишкою по компоненту;
- onKeyDown - коли при натисканні на кнопку на клавіатурі вона виявилася в нижньому положенні;
- onKeyUp - коли при відпуску клавіатурній кнопки вона виявилася в верхньому положенні;
- onKeyPress - виникає при натисканні на клавіатурну кнопку. Від подій onKeyDown і onKeyUp воно відрізняється типом використовуваного параметра Key ;

- `onMouseDown` - коли при натисканні кнопки мишки вона виявилася в нижньому положенні;
- `onMouseUp` - коли при відпуску кнопки мишки вона виявилася в верхньому положенні;
- `onMouseMove` - виникає при переміщенні покажчика мишки над компонентом;

3.2. База даних та програмування системи

Після отримання замовлення від клієнта інформація з приводу замовлення інформація перетворюється у стандартний формат для зберігання у базі даних під час побудови бази даних. Необхідно вибрати модель бази даних. Роблячи це потрібно врахувати наступне:

- Це повинна бути добре відома модель, щоб зробити заміну однієї реалізації бази даних іншою відносно простою.
- Це повинно дозволяти ефективні запити, що важливо, враховуючи, що вихідні моделі можуть бути досить великими.
- Він повинен підтримувати віддалений доступ до бази даних з одного або декількох географічно розподілених користувацьких інтерфейсів.
- Він підтримує злиття переглядів, поєднуючи інформацію з різних таблиць.
- Він підтримує мови запитів, які можуть виражати архітектурні шаблони.
- Контрольний пункт повинен підтримуватися реалізаціями, що означає, що проміжні результати можна зберегти. Це важливо в інтерактивному процесі, оскільки воно дає користувачеві свободу досліджувати з комфортом, розуміючи що зміни завжди можна скасувати.

Наприклад, робочий стіл Dali використовує реляційну модель бази даних. Він перетворює витягнуті подання (які можуть бути у багатьох різних форматах, залежно від інструментів, що використовуються для їх вилучення) у стандартну форму Rigi. Потім цей формат читається сценарієм perl і виводиться у форматі, що включає необхідний код SQL для побудови реляційних таблиць та заповнення їх вилученою інформацією. Рисунок 3.12 зображує схему цього процесу.

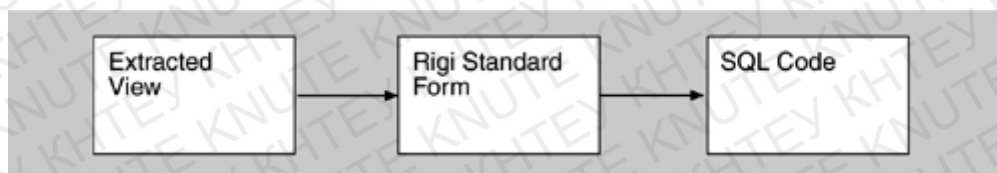


Рис. 3.12 Перетворення витягнутої інформації у формат SQL

Коли дані вводяться в базу даних, генеруються дві додаткові таблиці: елементи та взаємозв'язки. Вони перелічують витягнуті елементи та відносини відповідно.

Тут підхід до робочого середовища дозволяє застосувати нові інструменти та техніки, крім наявних на даний момент, для здійснення перетворення з будь-якого формату (форматів), який використовує інструмент вилучення. Наприклад, якщо інструмент потрібен для обробки нової мови, його можна створити, а його вихідні дані перетворити у формат робочого середовища.

У поточній версії робочого середовища Dali реляційна база даних POSTGRES забезпечує функціональність завдяки використанню SQL та perl для генерації та маніпулювання архітектурними. Сценарії SQL можна легко внести зміни, щоб зробити їх сумісними з іншими реалізаціями SQL.

Для своєї системи замовлень базу даних я вирішив використати Microsoft Access.

Microsoft Access — система управління базами даних, програма, що входить до складу пакету офісних програм Microsoft Office. Має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних. Це базова система яка є на кожній робочій станції оснащений операційною системою Windows, тому непотрібно встановлювати додаткову систему, але в разі необхідності RAD Studio дозволяє протягом десяти хвилин підключитися до Microsoft SQL Server без додаткових затрат. Але оскільки я не маю змоги запустити SQL Server то підключати свою систему я збираюся до Microsoft Access шлях до якої буду кожного разу вибирати власноруч.

Для підключення бази даних Microsoft Access до своєї системи замовлень я використав такі компоненти як:

- Компонент DBGrid - це візуальний компонент середовища Delphi, призначений для візуалізації наборів даних. Цей компонент знаходиться на сторінці компонентів DataControls палітри компонентів. Клас TDBCustGrid і клас TCustGrid - це батьківські класи компонента DBGrid. Компонент ділиться на стовпці (поля введення даних) і рядки (записи).
- DataSource - джерело даних, більше використовується для виведення інформації, що з таблиць, що з запитів наприклад в DBGrid.
- Компонент Delphi ADOConnection - забезпечує з'єднання інших компонентів ADO додатки з базою даних і здійснює управління транзакціями.
- Компонент Delphi ADOQuery - забезпечує застосування запитів SQL при роботі з даними через ADO. Це можуть бути запити перегляду даних SELECT і запити зміни даних INSERT, DELETE, UPDATE, ALTER TABLE, CREATE TABLE.

За допомогою цих компонентів я з'єднав систему з базою даних Microsoft Access.

Для кращої роботи з базою даних я зробив щоб система підключалася лише до таблиці яка має чітко визначену назву і незмінні назви стовпців для того щоб в системі відображалися дані лише чітко визначених для них місцях і був відсутній ризик перемішування даних в системі. Оскільки при прийомі і заповненні замовлення користуються посиланнями на чітко визначені стовпці в таблиці то ризик що назви стовпців таблиці або сама назва таблиці зміниться відсутня.

Оскільки система замовлень створюється лише для складських працівників то єдина таблиця яка їм може знадобитись це таблиця під назвою “Замовлення”, але сама таблиця для кращої адаптації має англійські позначення назв, тобто сама таблиця в базі даних Microsoft Access має назву “Orders”. Для цієї таблиці довелося використати SQL запит: `SELECT*FROM Orders`.

Для того щоб можна було кожного разу вибирати файл я створив форму на якій за допомогою компоненту OpenFileDialog, двох кнопок Button, поля введення виводу Edit та текстового поля Label створив можливість для користувача вибирати потрібну базу даних при необхідності.

Компонент OpenFileDialog не візуальний компонент призначений для підтримки операції відкриття файлів здатний працювати з будь-якими типами файлів. При зверненні до цього компоненту викликається стандартне діалогове вікно відкриття файлу (рис. 3.13).

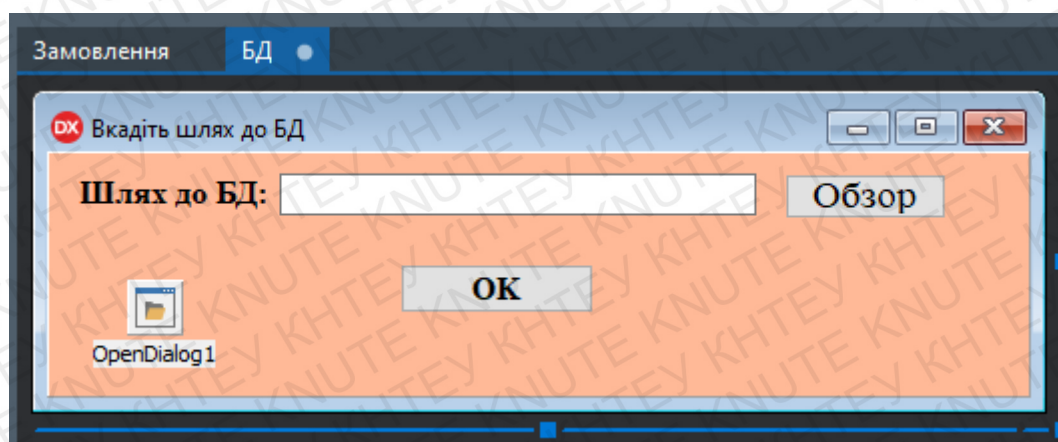


Рис. .3.13 Форма вибору бази даних

Для складського працівника потрібен такий елемент як друк накладної яка потрібна для кожного замовлення, тому на новій формі додавши компоненти TfrxDBDataset, TfrxReport для побудови файлу накладної, frxPDFExport, frxJPEGExport, frxDOCXExport, frxXMLExport, frxXLSEExport, це формати в яких користувач зможе зберегти накладну.

Тепер потрібно налаштувати TfrxReport, для цього лівою кнопкою миші двічі клацаємо по цьому компоненту. Відкривається вікно сходе на Microsoft Word в якому можна налаштувати вигляд накладної яка буде роздруковуватись (рис. 3.14).

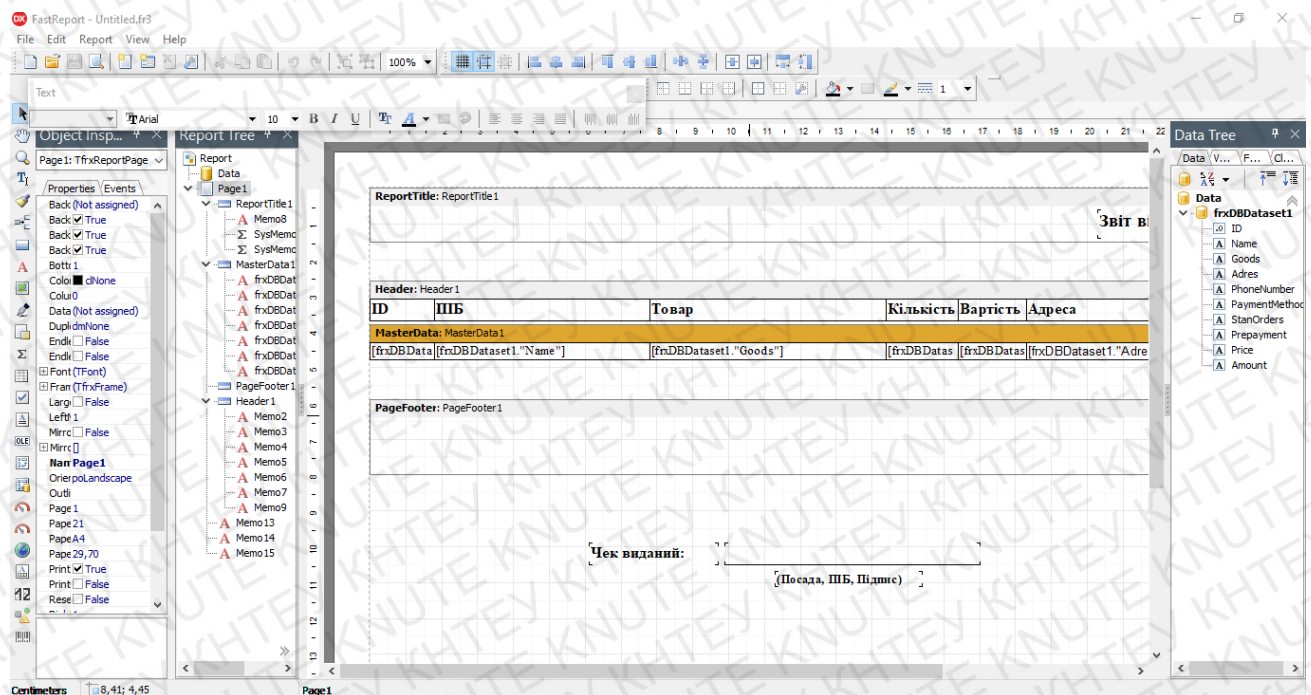


Рис. 3.14 Налаштування вигляду накладної

Тепер потрібно на першій формі додати компоненти для доступу до інших форм. Для цього на першу форму додамо компонент Компонент Delphi MainMenu призначений для додавання до програми головного меню, елемента, без якого не обходиться жодна з програм для Windows. Щоб додати до програми головне меню, потрібно розмістити на формі в довільному місці компонент MainMenu. Компонент MainMenu невізуальний, тобто, хоча і відображається на прототипі форми як невеликий квадрат, у чинній програмі не

буде видно. Опції головного меню створюються за допомогою спеціального редактора. Редактор меню викликається за допомогою подвійного клацання по компоненту MainMenu . В даному меню додаємо такі вкладки як “Файл” та “Вихід”. В вкладці “Файл” потрібно створити похідні вкладки:

- Перегляд звіту;
- Експорт звіту;
- Друк звіту;
- Відкрити базу даних;

За допомогою коду з’єднати “Перегляд звіту” з формою на якій знаходиться файл зі накладною, що зображена на рисунку 3.15.

Для вкладки “Експорт звіту” створюємо список, елементи якого будуть називатися:

- PDF file;
- XML file;
- DOCX file;
- XLS file;
- JPEG file;

Ці елементи за допомогою коду потрібно з’єднати з відповідними компонентами:

- FrxPDFExport;
- frxXMLExport;
- frxDOCXExport;
- frxXLSExport;
- frxJPEGExport;

Також щоб при випадковому натисканні на поле бази даних користувач не зміг нашкодити даним потрібно заборонити дозвіл на редагування через компонент DBGrid. Для цього в властивостях компонента DBGrid навпроти властивості ReadOnly встановити позначку True.

Але для того щоб користувач таки міг працювати з базою даних потрібно додати кнопки:

- Видалити;
- Додати;
- Редагувати;

Самі кнопки я створив за допомогою компоненту BitBtn. Кнопці видалити за допомогою коду надав можливість видаляти з таблиці виділене замовлення при натисканні на неї. І для кнопок під назвою “Додати” та “Редагувати” створив форми (див. рис. 3.16, рис. 3.17). На формах розмістив компоненти для введення Edit які підписав компонентами Label, також всі компоненти Edit під’єднав до певних стовпців таблиці бази даних, щоб мати змогу редагувати лише потрібний стовпець. Для збереження або відміни змін вирішив використати кнопки яким теж за допомогою коду дав можливість зберігати заповнені компоненти Edit в базу даних або цього не робити в залежності від функції кнопки.

DX

ПІВ DBEdit1

Товар DBEdit5

Адреса доставки DBEdit2

Номер телефону DBEdit3

Спосіб оплати DBEdit4

Стан замовлення DBComboBox1

Предоплата DBComboBox2

Ціна DBEdit6

Кількість DBEdit7

OK

Рис. 3.16 Форма редагування замовлення

The image shows a Windows application window titled "DX" with a standard title bar (minimize, maximize, close buttons). The window contains a form with the following fields and controls:

Field Label	Control Type
PIB	DBEdit1
Товар	DBEdit5
Адреса доставки	DBEdit2
Номер телефону	DBEdit3
Спосіб оплати	DBEdit4
Стан замовлення	DBComboBox1
Предоплата	DBComboBox2
Ціна	DBEdit6
Кількість	DBEdit7

At the bottom of the form, there are two buttons: "OK" and "Cancel".

Рис. 3.17 Форма додавання замовлення

Також потрібно якось додати можливість пошуку по таблиці. Щоб уникнути можливості пошуку по всім стовпцям таблиці якщо потрібно лише по одному над назвами стовпців я розмістив компоненти DBEdit в яких за допомогою коду прописав потрібні щоб DBEdit який знаходиться над стовпцем шукав лише по тому стовпцю над яким він розташований.

На цьому розробка системи замовлень завершена, можемо запускати програму.

3.3. Оцінка отриманих результатів проектування

Після завершення роботи над створенням системи замовлень потрібно обов'язково зберегти програму. Середовище програмування завжди зберігає проект в окрему папку яка має всі потрібні файли для запуску цього проекту на комп'ютері який не має самого середовища розробки. Також середовище програмування для кожного проекту завжди створює застосунок або по простішому “Ярлик”, за допомогою якого можна запустити створений проект. Сам застосунок можна винести на робочий стіл або закріпити на панелі задач. Тож натиснувши на застосунок відкривається створена система замовлень (рис. 3.18).

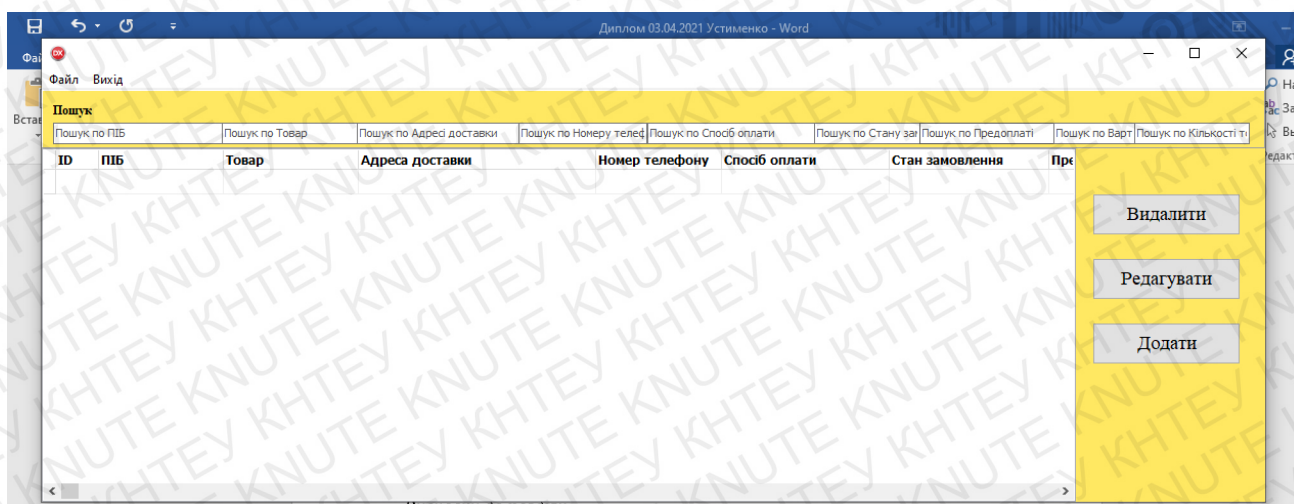


Рис. 3.18 Головна форма програми

Для початку потрібно вибрати базу даних, для цього в верхньому лівому куті знаходиться вкладка “Файл” в якій знаходиться додаткова вкладка “Відкрити БД” після натискання на яку відкриється форма в якій можна вибрати шлях до БД. Після відкриття бази даних на головному полі відобразяться дані які знаходяться в базі даних, це значить що підключення пройшло успішно (рис. 3.19).

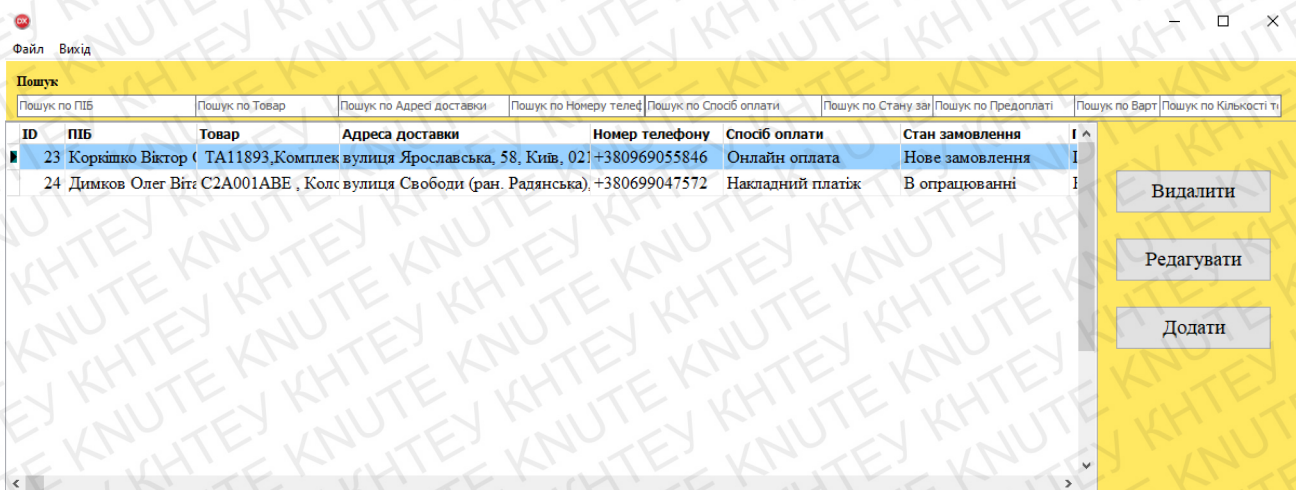


Рис. 3.19 Головна форма програми з відображеними в базі даних даними

Всі основні функції працюють, сортування проводиться чітко по визначених стовбцях, друк і експорт накладної працює як і планувалося. Система замовлень готова до впровадження її на підприємство.

ВИСНОВКИ

Нині доволі важко уявити життя без персонального комп'ютера. Паперові документи нам замінили електронні файли, замість одної фізичної книги ми носимо сотні книг в своєму телефоні. Індустріалізація і прогрес замінив фізичну працю машинною, Газети й журнали вийшли з моди. Жодна людина не зрівняється в роботі з роботом, електронні системи не знають втоми, вони не можуть помилитися виконуючи монотонну роботу. Системи які не мають власного інтелекту можуть працювати доки їхні апаратні деталі не вичерпають свій ліміт робочих годин, для них непотрібні заробітна плата або вихідні. Але через те що електронні системи не мають власного мислення вони досі повинні управлятись користувачами або операторами які зможуть в екстрених ситуаціях взяти все під свій контроль. По одинці людина і робот матимуть малий коефіцієнт корисної дії, але якщо дати їм працювати разом швидкість і якість роботи зросте в рази. Люди почали працювати з машинами ще в 18 столітті коли людей почали замінити машини, в 21 столітті рівень індустріалізації значно зріс, люди почали створювати автономні машини для важкої роботи, самі при цьому лише керували ними. І це дає свої плоди, рівень і тривалість життя порівняно з 18 століттям зріс і продовжує зростати завдяки тому що люди перестали наражати себе на небезпеку, також підприємці які максимально користуються новітніми технологіями і впроваджують їх в своє підприємство отримують значно більший прибуток порівняно з витратами ніж підприємці які досі всю роботу виконують фізично за рахунок більшого обороту замовлень та менших витрат на оплату праці. Наприклад швейний апарат в 10 разів швидше шиє ніж швачка, тобто ККД швейного апарату більше в 10 разів. На сучасному ринку йде змагання на виживання, той хто швидше і якісніше задовольнить клієнтів отримає прибуток. Бібліотеки, кінотеатри, комп'ютерні клуби закриваються оскільки на зміну їм прийшли онлайн сервіси для читання книг або перегляду фільмів які пропонують набагато кращий сервіс. Щоб уникнути закриття багато підприємств з застарілою системою встановлюють самі провідні технології але іноді здається що

ці технології здатен зрозуміти лише сам автор, не кажучи вже про працівників які тільки недавно освоїли телефон з сенсорним екраном. Тому перехід з паперових носіїв на електронні являється проблематичний, Саме для таких працівників я і створював систему замовлень, за допомогою моєї системи працівник який ніколи не працював комп'ютером а тим паче з комп'ютеризованою системою зможе освоїтись. Сподіваюся що за допомогою моєї системи я дам змогу працівникам які тільки почали працювати з програмним забезпеченням краще зрозуміти принцип роботи та скоріше вникнути в робочий процес.

Метою дипломного проекту була розробка системи замовлень.

В ході виконання дипломного проекту було:

- Виконано аналіз існуючих проблем в сфері обробки замовлень;
- Проведено огляд вже наявних на ринку систем та пошук їх відмінність та спільність.
- Розроблена система замовлень та її підключення до БД ;

Розроблена система дозволяє керувати замовленнями інтуїтивно, вона не навантажена непотрібними для нового користувача елементами, система має лише потрібний набір компонентів. Дана система дозволяє взяти на посаду складського працівника людину яка не має навичок в роботі з комп'ютеризованими системами обробки замовлень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. УКРІНФОРМ “В Україні рівень безробіття зріс майже до 10%” [Електронний ресурс] - Режим доступу: <https://www.ukrinform.ua/rubric-economy/3210897-v-ukraini-riven-bezrobitta-zris-majze-do-10-centr-razumkova.html>
2. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. [Електронний ресурс] - Режим доступу: <https://zakon.rada.gov.ua/rada/show/n0001217-96#Text>
3. “SQL Database Programming” (2015 Edition) By Chris Fehily” - <https://rk2bukz.cf/book.php?id=FRoDAQAAQBAJ>
4. Introducing Logistics Systems [Електронний ресурс] – Режим доступу: <https://catalogimages.wiley.com/images/db/pdf/0470849177.01.pdf>
5. ЛОГІСТИЧНІ СИСТЕМИ ТА ЇХ ЕЛЕМЕНТИ [Електронний ресурс] – Режим доступу: http://elar.khnu.km.ua/jspui/bitstream/123456789/4971/1/rozdil_3.pdf
6. Автоматизація системи управління замовленнями клієнтів для зростання вашого бізнесу [Електронний ресурс] – Режим доступу: <https://envybox.io/blog/avtomatizacija-sistemy-upravlenija-zakazami-klientov/>
7. Горшенина Е.В., Сорокин П.А. Информационные технологии в электронной коммерции. Оргтехника. – М.: Изд-во. центр "Академия", 2002. – 240 с. [Електронний ресурс] – Режим доступу: <https://cyberleninka.ru/article/n/internet-tehnologii-v-elektronnoy-kommertsii/viewer>
8. Досягнення цілей якості [Електронний ресурс] – Режим доступу: <https://people.ece.ubc.ca/matei/EECE417/BASS/ch13lev1sec5.html>
9. Кобилін А. М. Системи обробки економічної інформації : навчальний посібник / А. М. Кобилін . – Київ : Центр учбової літератури, 2019. – 234 с.
10. Чекіна, Д. Р. Система обробки замовлень в Інтернет магазині: дипломний проект бакалавра : 6.050103 Програмна інженерія / Чекіна Данило Романович. – Київ, 2019. – 112 с.

ДОДАТОК

Програмний код реалізації системи

Файл Замовлення.pas

unit ЗамоВлення;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.StdCtrls, Vcl.Buttons,
Vcl.Grids, Vcl.DBGrids, Data.Win.ADODB, Vcl.ComCtrls, Vcl.ExtCtrls, Vcl.Mask,
Vcl.DBCtrls, Vcl.Menus;

type

TForm1 = class(TForm)

ADOConnection1: TADOConnection;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

Panel1: TPanel;

MainMenu1: TMainMenu;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

PDFfile1: TMenuItem;

XMLfile1: TMenuItem;
DOCXfile1: TMenuItem;
XLSfile1: TMenuItem;
GPEGfile1: TMenuItem;
Edit1: TEdit;
Label1: TLabel;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
N7: TMenuItem;
Edit2: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure ВихідClick(Sender: TObject);
procedure PDFfile1Click(Sender: TObject);
procedure XMLfile1Click(Sender: TObject);
procedure DOCXfile1Click(Sender: TObject);
procedure XLSfile1Click(Sender: TObject);
procedure GPEGfile1Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);


```

procedure N7Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure Edit4Change(Sender: TObject);
procedure Edit5Change(Sender: TObject);
procedure Edit6Change(Sender: TObject);
procedure Edit7Change(Sender: TObject);
procedure Edit8Change(Sender: TObject);
procedure Edit9Change(Sender: TObject);
procedure Edit10Change(Sender: TObject);
procedure Panel1Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
implementation
  {$R *.dfm}

uses uЗамовлення, Редагування, Unit4, Звіт, БД;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form2.ShowModal;
  ADOQuery1.Insert;

```

```

end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    Form3.Show;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
    DBGrid1.DataSource.DataSet.Delete;
end;

procedure TForm1.DOCXfile1Click(Sender: TObject);
begin
    Form5.frxReport1.ShowReport();
    Form5.frxReport1.Export(Form5.frxDOCXExport1);
end;

procedure TForm1.Edit10Change(Sender: TObject);
begin
    if Form1.Edit10.Text="" then Form1.ADOQuery1.Filtered := False
    else
        begin
            Form1.ADOQuery1.Filtered := False;
            Form1.ADOQuery1.Filter:='Amount like '+form1.Edit10.Text+'%';
            Form1.ADOQuery1.Filtered := True;
        end;
end;
end;

```



```
procedure TForm1.Edit1Change(Sender: TObject);
begin
if Form1.Edit1.Text="" then Form1.ADOQuery1.Filtered := False
else
begin
Form1.ADOQuery1.Filtered := False;
Form1.ADOQuery1.Filter:='Adres like '+form1.Edit1.Text+'%';
Form1.ADOQuery1.Filtered := True;
end;
end;
```

```
procedure TForm1.Edit2Change(Sender: TObject);
begin
if Form1.Edit2.Text="" then Form1.ADOQuery1.Filtered := False
else
begin
Form1.ADOQuery1.Filtered := False;
Form1.ADOQuery1.Filter:='PhoneNumber like '+form1.Edit2.Text+'%';
Form1.ADOQuery1.Filtered := True;
end;
end;
```

```
procedure TForm1.Edit4Change(Sender: TObject);
begin
if Form1.Edit4.Text="" then Form1.ADOQuery1.Filtered := False
else
begin
Form1.ADOQuery1.Filtered := False;
Form1.ADOQuery1.Filter:='Name like '+form1.Edit4.Text+'%';
Form1.ADOQuery1.Filtered := True;
```

```
end;  
end;  
  
procedure TForm1.Edit5Change(Sender: TObject);  
begin  
if Form1.Edit5.Text="" then Form1.ADOQuery1.Filtered := False  
else  
begin  
Form1.ADOQuery1.Filtered := False;  
Form1.ADOQuery1.Filter:='Goods like '+form1.Edit5.Text+'%';  
Form1.ADOQuery1.Filtered := True;  
end;  
end;
```

```
procedure TForm1.Edit6Change(Sender: TObject);  
begin  
if Form1.Edit6.Text="" then Form1.ADOQuery1.Filtered := False  
else  
begin  
Form1.ADOQuery1.Filtered := False;  
Form1.ADOQuery1.Filter:='PaymentMethod like '+form1.Edit6.Text+'%';  
Form1.ADOQuery1.Filtered := True;  
end;  
end;
```

```
procedure TForm1.Edit7Change(Sender: TObject);  
begin  
if Form1.Edit7.Text="" then Form1.ADOQuery1.Filtered := False  
else
```



```
begin
  Form1.ADOQuery1.Filtered := False;
  Form1.ADOQuery1.Filter:='StanOrders like '+form1.Edit7.Text+'%';
  Form1.ADOQuery1.Filtered := True;
end;
end;
```

```
procedure TForm1.Edit8Change(Sender: TObject);
```

```
begin
  if Form1.Edit8.Text="" then Form1.ADOQuery1.Filtered := False
  else
    begin
      Form1.ADOQuery1.Filtered := False;
      Form1.ADOQuery1.Filter:='Prepayment like '+form1.Edit8.Text+'%';
      Form1.ADOQuery1.Filtered := True;
    end;
end;
```

```
procedure TForm1.Edit9Change(Sender: TObject);
```

```
begin
  if Form1.Edit9.Text="" then Form1.ADOQuery1.Filtered := False
  else
    begin
      Form1.ADOQuery1.Filtered := False;
      Form1.ADOQuery1.Filter:='Price like '+form1.Edit9.Text+'%';
      Form1.ADOQuery1.Filtered := True;
    end;
end;
```

```
procedure TForm1.GPEGfile1Click(Sender: TObject);
```

```
begin
```

```
Form5.frxReport1.ShowReport();
```

```
Form5.frxReport1.Export(Form5.frxJPEGExport1);
```

```
end;
```

```
procedure TForm1.N3Click(Sender: TObject);
```

```
begin
```

```
Form5.Show;
```

```
Form5.frxReport1.ShowReport();
```

```
end;
```

```
procedure TForm1.N5Click(Sender: TObject);
```

```
begin
```

```
Form5.frxReport1.ShowReport();
```

```
Form5.frxReport1.Print;
```

```
end;
```

```
procedure TForm1.N7Click(Sender: TObject);
```

```
begin
```

```
Form6.Show;
```

```
end;
```

```
procedure TForm1.PDFfile1Click(Sender: TObject);
```

```
begin
```

```
Form5.frxReport1.ShowReport();
```

```
Form5.frxReport1.Export(Form5.frxPDFExport1);
```

```
end;
```



```
procedure TForm1.XLSfile1Click(Sender: TObject);
```

```
begin
```

```
Form5.frxReport1.ShowReport();
```

```
Form5.frxReport1.Export(Form5.frxXLSExport1);
```

```
end;
```

```
procedure TForm1.XMLfile1Click(Sender: TObject);
```

```
begin
```

```
Form5.frxReport1.ShowReport();
```

```
Form5.frxReport1.Export(Form5.frxXMLExport1);
```

```
end;
```

```
procedure TForm1.ВихідClick(Sender: TObject);
```

```
begin
```

```
Form1.Close;
```

```
end;
```

```
end.
```