

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизація і комп'ютерних систем

Кафедра Інформаційних систем

Освітній ступінь Бакалавр

Спеціальність 122 «Комп'ютерні науки та інформаційні технології»
(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»
(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інформаційних систем

С.М.Чумаченко

“___” _____ 20__ року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Устимуку Івану Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Створення системи підпримки прийняття рішень логіста підприємства з вантажних перевезень»

керівник роботи Грибков Сергій Віталійович, доцент, к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2020 року №269-кв

2. Строк подання здобувачем роботи 29.05.2020 року

3. Вихідні дані до роботи посадові інструкції, статут підприємства, інформація про угоди, замовлення та технічний парк підприємства, зразки документів

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) системний аналіз діяльності підприємства, функціональний аналіз виконання замовлень на підприємстві, постановка задачі, створення логічної та фізичної моделей даних, бази даних, створення СППР, , інструкція користувача, заходи з охорони праці, висновки

5. Перелік графічного матеріалу

функціональна модель «AS-IS» процесу виконання замовлень, функціональна модель «TO-BE» процесу виконання замовлень, моделі бази даних, структура бази даних, інтерфейс користувача

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ	Грибков С.В.		
1	Грибков С.В.		
2	Грибков С.В.		
3	Грибков С.В.		

7. Дата видачі завдання _____ 27 квітня 2020 року _____

КАЛЕНДАРНИЙ ПЛАН

№ З№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Формування вступу	29.04.2020	виконано
2	Системний аналіз діяльності підприємства	01.05.2020	виконано
3	Функціональне моделювання	04.05.2020	виконано
4	Пошук та аналіз ІС	07.05.2020	виконано
5	Розробка логічної та фізичної моделей даних	08.05.2020	виконано
6	Генерація бази даних в СУБД	10.05.2020	виконано
7	Розробка елементів системи	12.05.2020	виконано
8	Створення інструкції користувача	14.05.2020	виконано
9	Розробка розділу з охорони праці	23.05.2020	виконано
10	Завершення роботи над пояснювальною запискою	26.05.2020	виконано
11	Створення презентації	28.05.2020	виконано

Здобувач _____
(підпис)

Устимук І.В. _____
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Грибков С.В. _____
(прізвище та ініціали)

АНОТАЦІЯ

Прізвище та ініціали здобувача: Устимук І.В.

Назва кваліфікаційної роботи: Створення системи підтримки прийняття рішень логіста підприємства з вантажних перевезень

Кваліфікаційна робота містить: 90 сторінку, 3 розділи, 8 таблиці, 5 додатки, 14 джерел.

Спеціальність: 122 «Комп'ютерні науки та інформаційні технології»

Метою кваліфікаційної роботи є розробка системи підтримки прийняття рішення логіста ФОП «Устимук В.С.». Головною задачею системи є збільшення продуктивності та зменшення часових витрат на обробку замовлень логістами підприємства. Метою даної роботи є створення системи, за допомогою якої логіст зможе отримувати всі необхідні для прийняття рішення йому дані.

Кваліфікаційна робота базується на знаннях отриманих під час навчання в університеті та під час проходження практик (виробничої та переддипломної). У кваліфікаційній роботі подана загальна характеристика об'єкту автоматизації, а саме ФОП «Устимук В.С.», сформовані основні вимоги до створеної системи, визначені функції, що автоматизувалися, обґрунтовано розробку інформаційної системи, подано аналіз ефективності впровадження ІС. Створена інформаційна система забезпечить підвищення ефективності роботи логіста даного підприємства, та роботу підприємства в цілому.

Ключові слова: ФІЗИЧНА ОСОБА ПІДПРИЄМЕЦЬ, ЛОГІСТИКА, ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРФЕЙС, ФУНКЦІОНАЛЬНА МОДЕЛЬ, БАЗА ДАНИХ, СУБД, SQL SERVER, VISUAL STUDIO.

THE SUMMERY

Surname and initials of the applicant: Ustymuk I.V.

Title of qualification work: Development of a decision support system for the logistics of freight company

Qualification work contains: 90 pages, 3 sections, 8 tables, 5 appendices, 14 sources.

Specialty: 122 "Computer Science and Information Technology"

The purpose of the qualification work is to develop a decision support system for the logistics of IE "Ustymuk VS". The main task of the system is to increase productivity and reduce time spent on processing orders by logistics companies. The purpose of this work is to create a system through which the logistician will be able to obtain all the necessary data for decision-making.

Qualification work is based on the knowledge gained during study at the university and during internships (industrial and undergraduate). The qualification work presents the general characteristics of the object of automation, namely PPE "Ustymuk VS", formed the basic requirements for the system, identified the functions that are automated, substantiated the development of information system, analyzes the effectiveness of IS implementation. The created information system will increase the efficiency of the logistics of the enterprise, and the work of the enterprise as a whole.

Keywords: INDIVIDUAL ENTREPRENEUR, LOGISTICS, INFORMATION SYSTEM, INTERFACE, FUNCTIONAL MODEL, DATABASE, DBMS, SQL SERVER, VISUAL STUDIO.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ФОП “УСТИМУК В.С.” ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ	9
1.1. Загальна характеристика ФОП “Устимук В.С.”	9
1.2. Організаційна структура ФОП “Устимук В.С.”	11
1.2.1. Загальна схема організаційної структури підприємства	11
1.2.2. Схема структури логістичного відділу підприємства	11
1.2.3. Взаємодія логістичного відділу з іншими відділами підприємства 13	
1.3. Аналіз нинішнього стану автоматизації	14
1.4. Функціональне моделювання існуючих бізнес-процесів	14
1.4.1. Обґрунтування загального опису стандарту IDEF0	14
1.4.2. Розроблення функціональної моделі існуючих бізнес-процесів “як є” ФОП “Устимук В.С.”	15
1.4.3. Виявлені проблеми	17
1.5. Аналіз існуючих аналогів ІС для підтримки логістів	18
1.5.1. 1С: Підприємство	18
1.5.2. Bookkeeper Saas	19
1.5.3. Дебет Плюс	20
1.5.4. Своя технологія	21
1.5.5. Openbravo	22
1.5.6. Результат порівняння систем-аналогів	23
1.6. Обґрунтування доцільності проектування й розроблення системи підтримки прийняття рішень логіста	23
1.7. Концептуальна модель створюваної системи	24
1.8. Постановка задачі	25
1.8.1. Призначення та мета створення системи	25
1.8.2. Вимоги до створюваної системи	25
1.8.3. Функції які повинна виконувати система	26
РОЗДІЛ 2. РОЗРОБКА СППР	29
2.1.1. Використані програмні засоби	29

2.1.2. Використані технології.....	29
2.2. Проектування БД	30
2.3.Розробка інтерфейсу користувача.....	34
2.4. Інструкція користувача.....	54
2.4.1. Вхід до системи.....	55
2.4.2. Головне вікно програми	56
2.4.3. Робота з довідниками	57
2.4.4. Робота з пошуками.....	59
2.4.5. Робота зі звітами	60
2.5. Техніко-економічне обґрунтування	60
2.5.1. Розрахунок оплати праці.....	60
2.5.2. Визначення чисельності виконавців	63
2.5.3. Витрати, пов'язані з розробкою інформаційної системи:	64
2.5.4. Загальна вартість розробки і впровадження системи.....	65
РОЗДІЛ 3. ОХОРОНА ПРАЦІ	67
3.1. Аналіз умов праці.....	67
3.2. Характеристика та організація робочого місця	67
3.3. Вимоги безпеки праці при роботі з комп'ютером	70
3.4. Пожежна безпека приміщення.....	72
ВИСНОВОК.....	74
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
ДОДАТКИ.....	77

ВСТУП

Ми живемо в дивовижний час: коли наука та технології розвиваються не щоднини, а щогодини, і прості студенти, маючи за плечами лише ідею та віру в неї, здатні побудувати свою багатомільярдну імперію в гаражі.

Здається, в 2020 році все що можна було, вже давно автоматизовано, але кожна компанія, кожен бізнес-процес по своєму унікальний і не завжди шаблонні рішення здатні задовільнити бажання цих компаній. І, на мою думку, всі компанії: і великі, і малі бажають вони цього чи ні, обов'язково пройдуть “діджиталізацію”.

Актуальність даної роботи полягає в тому, що розробка індивідуального ПЗ під обраний бізнес-процес можливо потребує більше ресурсів, як часових так і фінансових, але набагато чіткіше та в більшій мірі здатне задовільнити потреби обраного бізнес-процесу, ніж існуючі на ринку, програми для масового використання.

Об'єктом дослідження моєї кваліфікаційної роботи є ФОП “Устимук В.С.”.

Предмет дослідження — організації роботи логістичного відділу ФОП “Устимук В.С.” та напрями її вдосконалення.

Метою роботи є дослідження функцій логістичного відділу ФОП “Устимук В.С.” та розробка системи допомоги прийняття рішень логіста.

Складність роботи логіста, полягає у складності прийняття рішення при обранні замовлення та того, хто буде виконувати. Він повинен швидко розрахувати економічну вигідність замовлення, для того, щоб інші перевізники не прийняли це замовлення першими, а на даний момент він робить це вручну, що є не дуже швидким рішенням, яке уповільнює роботу.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ФОП “УСТИМУК В.С.” ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ

1.1. Загальна характеристика ФОП “Устимук В.С.”

ФОП “Устимук В.С.”— це підприємство, яке здійснює вантажні перевезення по Україні, доставляє негабаритні і збірні вантажі на замовлення інших юридичних або фізичних осіб.

Основним способом пошуку замовлень компанією являються Інтернет ресурси, такі як наприклад lardi-trans.com.

Lardi-trans — це інформаційний ресурс, основна функція якого направлена на забезпечення оперативною інформацією про актуальні пропозиції щодо перевезення вантажів і вільних транспортних засобів всіх учасників транспортного ринка (замовників, перевізників та експедиторських компаній)[1].

Звичайно після n-го виконання замовлення для одного підприємства формуються певні зв'язки між компаніями і, в подальшому, компанія-замовник напряму звертається до обраної компанії-виконавця для виконання замовлення, так формується другий шлях отримання замовлень компанією: прямі звернення компанії-замовника.

Матеріально-технічна база досліджуваного підприємства — сукупність усіх матеріальних умов здійснення процесу виконання вантажних перевезень в поєднанні з алгоритмом вибору економічно вигідних замовлень.

Найважливішою складовою матеріально-технічної бази в цьому випадку є автомобілі — головний засіб виконання замовлення підприємством.

Усі автомобілі поділяються на дві групи: великі та маленькі.

До маленьких будуть відноситися міні-фургони, буси, та тентовані буси, них також називають “пландека”, що представлені на рисунку 1.1.



Рис.1.1.Пландека

А до великих відносяться так звані фури, або напівпричепи-вантажівки. Напівпричіп-вантажівка — це комбінація тягача та одного або декількох напівпричепів для перевезення вантажів.

Для різних замовлень використовуються різні автомобілі, це зумовлюється тим, що кожне замовлення, якщо це не постійне замовлення, має свою власну унікальну вагу та об'єм, і це якщо не враховувати, що об'єм вираховується по формулі в якій присутні висота, ширина та довжина. Всі ці умови накладають певні обмеження на мінімальний розмір авто.

В цьому аспекті пландека має перевагу перед іншими представниками групи маленьких машин через збільшений об'єм, але для них використовують важкі каркаси для збільшення площі, які можуть зменшити вантажопідйомність.

Необхідно відмітити, якщо буде велика кількість фур, які будуть їздити майже порожні принесе підприємству більше збитків ніж прибутків.

Для кожного автомобіля потрібен свій керманіч, що також визначає певні обмеження на перевезення певних грузів.

Також необхідно відмітити вплив людського фактору на матеріально-технічну базу, що полягає у несвоєчасному опрацюванні заявок на перевезення менеджером або ж помилки при розрахунку економічної вигідності замовлення логістом.

1.2. Організаційна структура ФОП “Устимук В.С.”

1.2.1. Загальна схема організаційної структури підприємства

Організаційно-функціональна структура підприємства складається з 4 відділів, якими керує генеральний директор, її вона відображена на рисунку 1.2.



Рис.1.2. Загальна структура підприємства

1.2.2. Схема структури логістичного відділу підприємства

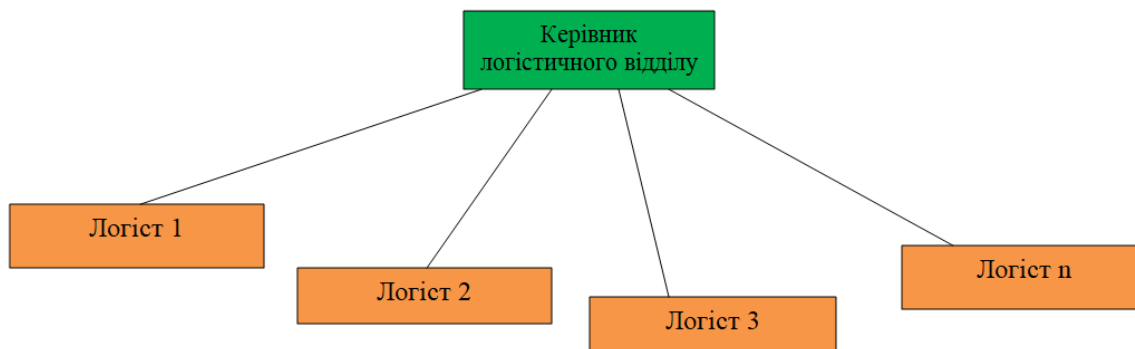


Рис. 1.3. Схема структури логістичного відділу

Логістичний відділ виконує на підприємстві дуже важливу роль – здійснює оцінку економічної вигідності замовлення.

Кожне замовлення має оголошену ціну, яку підприємство отримає за його виконання, якщо підприємство буде виконувати всі замовлення підряд, не враховуючи економічну вигідність, то дуже швидко підприємство припинить своє існування, адже кожен замовник хоче чим більше заощадити на перевезенні свого вантажу, а конкуренція між перевізниками в останні роки посилилася через збільшення їх кількості.

При цьому недосвідчені перевізники дуже часто погоджуються на виконання замовлення за доволі низькими, по міркам досвідчених перевізників цінам. Вони отримують прибуток, але не значний, при умові виникнення якоїсь несправності в автомобіля або інших непередбачених обставин, сума витрачена на усунення несправності повертається набагато повільніше ніж при виконанні замовлень за вищу ціну. В свою чергу ось такі дії зі сторони недосвідчених перевізників утворюють на транспортному ринку ситуацію, при якій ціни на всі замовлення починають зменшуватися.

Також логістичний відділ обирає, яке авто буде виконувати те, чи інше замовлення.

Функціональну схему роботи підрозділу представлена на рисунку 1.4.

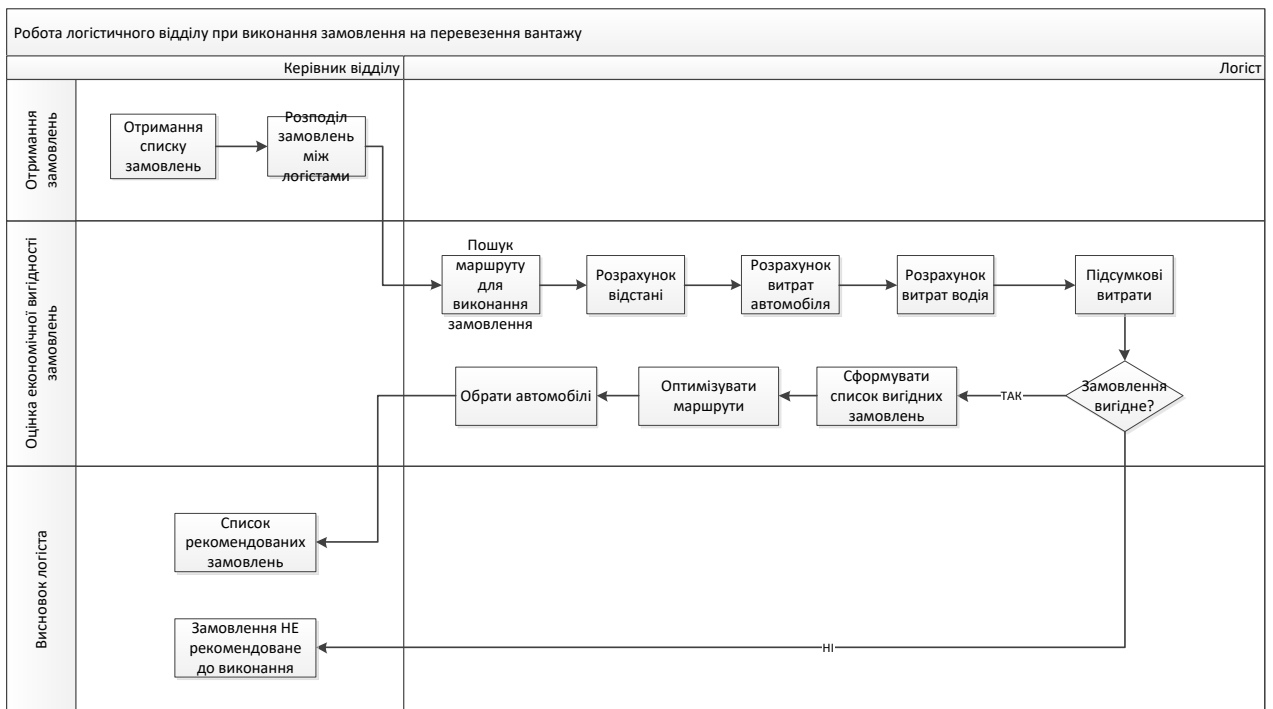


Рис.1.4. Функціональна схема роботи логістичного відділу

1.2.3. Взаємодія логістичного відділу з іншими відділами підприємства

Взаємодія логістичного відділу з іншими відділами підприємства представлена в таблиці 1.1.

Таблиця 1.1. Взаємодія логістичного відділу з іншими відділами підприємства

№	Відділ	Взаємодія
1	Менеджери	Менеджери передають отримані замовлення до логістичного відділу для пошуку вигідних для компанії замовлень; Логістичний відділ повертає інформацію про замовлення, з врахованою економічною вигідністю замовлень, а також переліком авто для виконання замовлень.
2	Бухгалтерія	Нараховує зарплату для всіх співробітників логістичного відділу;

		Логістичний відділ передає інформацію про ціну на замовлення бухгалтерам; Розраховує вартість перевезення; Формує квитанцію до сплати.
4	Водії	Водії отримують від логістів план виконання замовлення та маршрути для виконання замовлення.

1.3. Аналіз нинішнього стану автоматизації

Стан автоматизації ФОП “Устимук В.С.” на сьогоднішній день являється дуже низьким.

Не дивлячись на те, що пошук замовлень, розрахунок економічної вигідності, формування угод, їх відправка замовнику виконується за участі комп’ютера при цьому на підприємстві не використовується ніяка ІС.

Розрахунок економічної вигідності замовлення виконується логістами вручну, хоча при цьому також використовується ПК.

Для формування угод менеджери використовують ПЗ Microsoft Word, але так як в Україні відсутній електронний підпис, їх доводиться роздруковувати для підпису, після чого робити фото або скановану копію та надсилати електронною поштою.

Ведеться облік угод, але він ведеться у паперовій формі, а це значно уповільнює отримання інформації про сплату за угодою, яка використовується вкінці кожного місяця для сплати податків.

1.4. Функціональне моделювання існуючих бізнес-процесів

1.4.1. Обґрунтування Загальний опис стандарту IDEF0

Функціональна модель існуючих бізнес-процесів розроблюється за допомогою стандарту IDEF0, тому що ця методологія графічного опису систем і процесів діяльності організації як безлічі взаємозалежних функцій.

Вона дозволяє досліджувати функції організації, не пов'язуючи їх з об'єктами, що забезпечують їх реалізацію.

У стандарті IDEF0 за допомогою входу показують об'єкти – інформаційні та матеріальні потоки, які перетворюються в бізнес-процесі.

За допомогою управління показуються об'єкти – матеріальні та інформаційні потоки, які перетворюються на процесі, та потрібні для його виконання.

Використовуючи механізми IDEF0 можна відображати інструменти та ресурси, за допомогою яких бізнес-процес реалізується (наприклад: технічні засоби, люди, інформаційні системи і т.д.). Вихід бізнес-процесу, описаного в стандарті IDEF0, повністю відповідає за змістом виходу процесу, описаного за допомогою DFD-схеми[2].

1.4.2. Розроблення функціональної моделі існуючих бізнес процесів “як є” ФОП “Устимук В.С.”

Функціональна модель в стандарті IDEF0, створена за допомогою програмного засобу AllFusion Process Modeler. Розроблена мною функціональна модель описує процеси необхідні для пошуку та виконання замовлення підприємством в тому вигляді, як вони здійснюються зараз (AS-IS).

Функціональна модель складається з 4 рівнів. Позначення рівнів починається з верхнього, який позначається A0. Наступний рівень декомпозиції позначається A1-Am, де m – номер блоку. На кожному наступному рівні декомпозиції, частини діаграми отримують своє позначення на основі номера батьківського блоку плюс порядковий номер на поточному рівні, тобто, A.3.2, означає, що це другий блок в третій діаграмі.

Таким чином, формується набір взаємозв'язаних діаграм, що відображає всі процеси, що відбуваються під час вибору та виконання замовлення(Додаток А).

На рисунку 1 додатку А зображений верхній рівень діаграми функціональної моделі.

На верхньому рівні ми можемо побачити всі механізми керування та управління, а також вхідні та вихідні дані.

Вхідними даними для нашого процесу є:

- замовлення на перевезення вантажу;
- вантаж;
- завдання на пошук замовлень.

Вихідні дані:

- акт виконання замовлення;
- доставлений вантаж;
- квитанція про сплату за замовлення.

Керування:

- менеджер;
- ПК;
- логіст;
- водій;
- бухгалтер.

Управління:

- Женевська конвенція;
- внутрішні правила підприємства.

На першому рівні декомпозиції (Рис.2, Додаток А) зображені основні бізнес-функції, що необхідні для виконання нашого бізнес-процесу, а також їх послідовність та взаємозв'язок.

В результаті аналізу функціональної моделі виділяємо вузьке місце, а саме “Вибір замовлень для виконання”, за яку відповідає відділ логістики..

Виділена бізнес-функція більш детально розглядається на другому рівні декомпозиції (Рис.3, Додаток А).

На цій діаграмі ми можемо розглянути як на даний момент виглядає процес вибору замовлень логістами, а також на Рис.4, Додатка А можна побачити декомпозиції пункту “Обрати авто”.

Після розгляду всіх діаграм можна зробити висновки стосовно наявних проблем, а на основі наявних проблем сформуванати перелік задач автоматизації.

1.4.3. Виявлені проблеми

На основі аналізу функціональної моделі були виявлені наступні проблеми:

- Недостатній рівень автоматизації;
- Відсутність БД підприємства;
- Неможливість оперативно контролювати виконання замовлень;
- Паперовий документообіг;
- Підготовка звітності займає багато часу;
- Логісти витрачають багато часу на економічну оцінку замовлень та прийняття рішень при їх обрання.

Для того, щоб усунути перераховані вище проблеми, необхідно створити систему підтримку прийняття рішень для логістів та менеджерів.

1.5. Аналіз існуючих аналогів ІС для підтримки логістів

1.5.1. 1С: Підприємство



Рис. 1.5. 1С:Підприємство

1С:Підприємство — має широке застосування та поширене у країнах СНГ. Останнім часом на території України заборонено для використання у державних установах, адже центральний офіс та саме програмне забезпечення відноситься до Росії, що є країною загарбником та веде не припустимі дію проти України.

Представники середнього та малого бізнесу використовують його для ведення різних обліків на підприємстві. Особливістю даної системи є те що воно має досить розширений функціонал та невелику вартість.

Має модулі бухгалтерського і податкового обліку, які враховують особливості усіх країн в яких вона використовується.

Розробники займають серйозну нішу на ринку програмного забезпечення, адже має досить розвинену систему технічної підтримки від 500 фірм партнерів на території України [3].

1.5.2. Bookkeeper SaaS

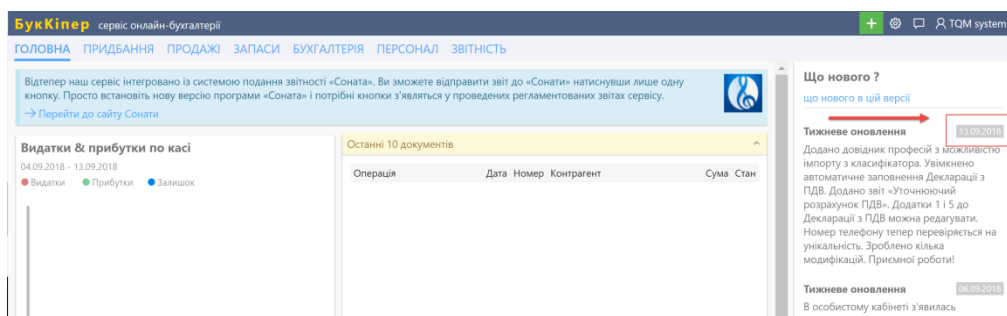


Рис. 1.6. Bookkeeper SaaS

Bookkeeper SaaS - це web-сервіс для автоматизації бухгалтерського обліку. Розробником є українська фірма. Перший місяць роботи безкоштовний, потім 295 грн. в місяць.

Ця ІС постійно вдосконалюється та розвивається.

Стандартний функціонал полягає у підтримки наступної діяльності:

- закупівля;
- продаж;
- управління запасами;
- бухгалтерія;
- управління персоналом;
- усе що пов'язано з ПДВ;
- формування бухгалтерських звітностей;
- формування та відправлення податкових звітів.

Як переконує фірма розробник на офіційному сайті – веб-інтерфейс Bookkeeper SaaS може працювати в усіх сучасних браузерях.

Обмеження на її використання це застарілі версії операційних систем та браузерів. Бази даних мають високу надійність і безпеку, адже система орієнтована на використання на СУБД Microsoft, що забезпечує високу надійність і безпеку[4].

1.5.3. Дебет Плюс

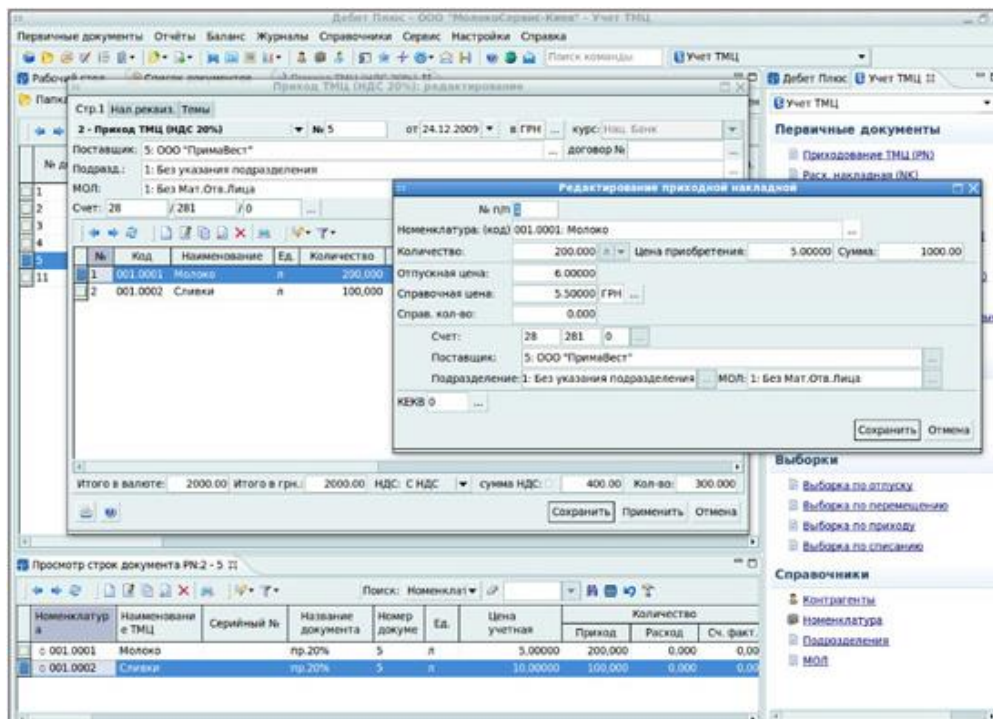


Рис. 1.7. Дебет Плюс

Безкоштовну версію Дебет Плюс можуть використовувати як підприємці, так і малі підприємства. Система підтримує ведення складського обліку, а також орієнтована на взаємодію з клієнтами підприємства. У функції системи входить звіт бухгалтерського балансу, облік основних засобів, заробітна плата. Програма поставляється під ОС Windows, Linux, Mac OS.

Безкоштовна версія системи "Дебет Плюс":

- підходить як для підприємців, так і для малих підприємств;
- дозволяє вести складський облік як з бухгалтерією, так і без бухгалтерії (за бажанням користувача);
- Працює на різних ОС - Windows, Linux, Mac OS, і СУБД - PostgreSQL, MySQL;
- повністю готова для роботи в умовах законодавства України і оперативно оновлюється в зв'язку з його змінами[4].

1.5.4. Своя технологія

«Своя технологія» - платформа направлена на підтримку управлінського обліку. Основна конфігурація системи, що забезпечує основні вимоги більшості українських фірм, поширюється безкоштовно.

Безкоштовна версія може бути використана в комерційних організаціях з дотриманням усіх ліцензійних вимог.

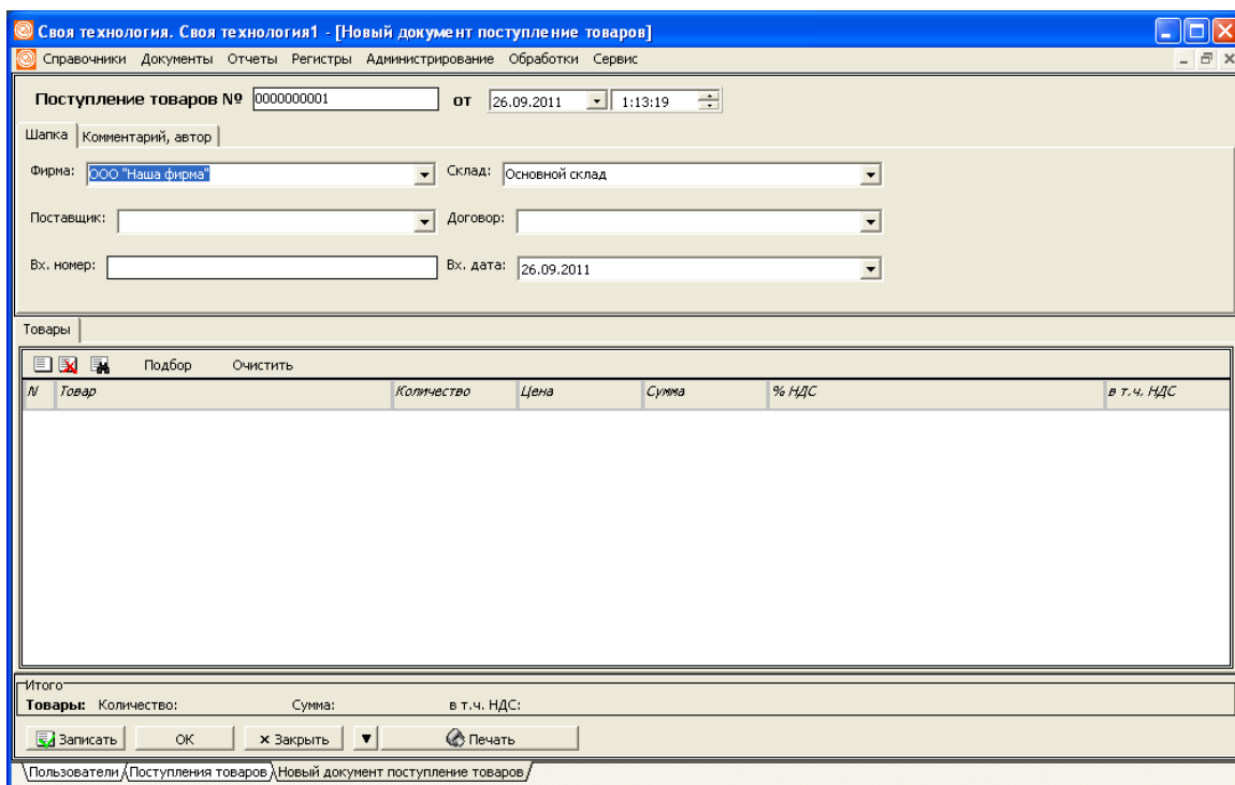


Рис. 1.8. Своя технологія

Основні можливості даної ІС:

- облік продукції, товарів, матеріалів;
- облік продажу товарів;
- аналіз продажу товарів;
- ведення взаєморозрахунків з клієнтами;
- облік руху грошових коштів ;
- оформлення усіх первинних документів.

За всіма напрямками обліку є можливість формувати відомості, а також є можливість формування додаткових звітів. Звіти та друковані форми можуть бути експортовані у Excel, Open Office. Усі сформовані звіти можливо відправити на електронну пошту. Система дозволяє вивантажити інформацію у ІС бухгалтерського обліку (при використанні відповідних додаткових модулів) [4].

Переваги даної ІС [4]:

- висока швидкість та надійність роботи;
- використання водночас великою кількістю користувачів однієї бази даних, що майже унікально для ІС, які поширюються безкоштовно;
- можливість роботи з декількома базами даних;
- формування різноманітних звітів;
- можливість підключення та використання через RS232 або USB різного торгового устаткування;
- автоматичне виправлення помилок.

1.5.5. Openbravo

Openbravo Безкоштовна ERP система з відкритим кодом, що забезпечує підтримку роботи наступних напрямків:

- наскрізний фінансовий облік;
- продажі і CRM;
- закупівлі;
- склад;
- виробництво;
- управління проектами та обслуговуванням.

Перевагою є те, що вона є WEB орієнтованою та забезпечує прості зміни та оновлення, гнучка в розгортанні, адже легко працює на Windows та Linux. Система забезпечує високу віддачу при мінімальній вартості

Мінімальні початкові інвестиції, адже підтримує модель "Плати тільки за послуги" знижує інвестиційне навантаження і дозволяє чітко контролювати вартість отриманого результату[4].

1.5.6. Результат порівняння систем-аналогів

Усі розглянуті системи являються повноцінними додатками або web-ресурсами, для ведення управління різними функціями підприємства, а саме фінансовим обліком, складським управлінням, управління виробничим контуром тощо. Використовуючи одну з цих систем є можливість задовольнити потреби переважної більшості невеликих підприємств або підприємців.

Проте, такий потужний та широкий функціонал має і свої недоліки. Персоналу потрібно більше практики для опанування програмного засобу, а широкий функціонал потребує більше обчислювальної потужності для стабільного функціонування.

Також деякі з цих програм мають безкоштовні версії, а платні, мають ціну, нижчу за ціну європейських програм подібного рівня, оскільки більшість орієнтована на користувачів з пострадянського простору, де доходи підприємств нижче європейських, а отже і сума, яку підприємство готове віддати за будь-який продукт буде меншою.

1.6. Обґрунтування доцільності проектування й розроблення системи підтримки прийняття рішень логіста

На підприємстві ФОП "Устимук В.С." немає інформаційної системи для підтримки роботи відділу логістики. На основі результату порівняння існуючих на ринку інформаційних систем-аналогів видно, що вони не цілком

задовольняють потреби відділу логістики, що виявлені під час функціонального моделювання предметної області. Необхідно відмітити, що більшість систем є дуже громіздкі, та мають велику кількість непотрібних функцій, які будуть ускладнювати користування даними системами. Враховуючи усе вище сказане, актуальним завданням є розробка власної інформаційної системи підтримки прийняття рішень логіста для ФОП “Устимук В.С.”.

1.7. Концептуальна модель створюваної системи

Основною відмінністю концептуальної моделі системи (TO-BE) від моделі AS-IS в тому що тут з’явився новий механізм, а саме інформаційна система, що буде створена та використовуватися, це можна побачити в додатку Б на рисунку 1.

Основні зміни можна побачити на рисунку 2 додатку Б. Адже тепер весь процес вибору економічно вигідних замовлень зводиться до роботи логіста в інформаційній системі, там є вся необхідна інформація, а також механізми, які спрощують виконання розрахунку.

Також на рисунку 3 додатку Б можна побачити процес вибору автомобіля для виконання замовлення, цей процес тепер також виконується в ІС, а облік автомобілів в БД набагато пришвидшує цей процес.

На даний момент, розроблювана ІС буде являти собою систему підтримки прийняття рішень логіста, проте в подальшому розвитку, вона може перетворитися у корпоративну інформаційну систему всього підприємства.

При такому масштабуванні, вже обов’язковим буде механізм входу до системи з різними повноваженнями в залежності від того хто входить до системи, для того аби забезпечити різних працівників, різними повноваженнями в залежності від їх потреб.

В майбутньому, можливо розробити модуль, який буде вираховувати відстань автоматично, за допомогою Google maps та Google API для програмного доступу до них.

1.8. Постановка задачі

1.8.1. Призначення та мета створення системи

Створювана система буде забезпечувати підтримку прийняття рішень логістів підприємства при обранні замовлення та їх економічної оцінки замовлень, формування списку вигідних замовлень, та вибору автомобілів для виконання замовлень. До основних цілей інформаційної системи відноситься зберігання інформації про автомобілі, водіїв, замовників, замовлення та угоди підприємства, та забезпечення швидкого доступу до цієї інформації для логістів підприємства, це значно зменшить час, який буде витрачати логіст на обробку нових замовлень.

1.8.2. Вимоги до створюваної системи

- Процесор: Intel Pentium 4 або AMD Athlon XP чи вище
- Оперативна пам'ять: 512 МБ або більше
- Відеокарта: зі 128 або більше МБ відеопам'яті
- ОС: Windows 7/8/8.1/10
- СУБД: MS SQL Server 2008
- Наявність мишки та клавіатури
- Системи повинна виконувати резервне копіюванні при роботі з даними
- Перед використанням системи, персонал обов'язково повинен ознайомитися з інструкцією користувача.

1.8.3. Функції які повинна виконувати система

Для підтримки прийняття рішень в роботі логіста, система повинна забезпечувати наступні пошуки:

- пошук всіх замовлень замовника;
- пошук всіх угод замовника;
- пошук автомобілів заданої вантажопідйомності або вищої;
- пошук інформації про водія та автомобіль за номером автомобіля;
- пошук необроблених замовлень;
- пошук вільних автомобілів;
- пошук угоди за її номером;
- пошук акту виконаних робіт та квитанції для обраної угоди;
- пошук мінімальної вартості необроблених замовлень(на основі відстані).

Для взаємодії логістів з БД, розроблений додаток повинен забезпечити наступні функції:

- Перегляд записів в БД;
- Редагування записів в БД;
- Додавання записів в БД;
- Видалення записів в БД.

Для спрощення ведення звітності, система повинна забезпечувати автоматичне формування наступних звітів:

- угоди по замовникам;
- замовлення по замовникам;
- угоди по автомобілях;
- угоди за обраний місяць.

Функції, які необхідна забезпечити система а також вхідні та вихідні дані для них наведені в таблиці 1.2.

Таблиця 1.2. Функції, які необхідна забезпечити система

№	Найменування функції	Вхідна інформація	Вихідна інформація
1	Перегляд записів	Таблиці БД	Форми зі списками
2	Редагування записів	Форми зі списками	Відредаговані таблиці БД
3	Видалення записів	Форми зі списками	Відредаговані таблиці БД
4	Пошук всіх замовлень замовника	Назва замовника	Форма зі списком замовлень
5	Пошук всіх угод замовника	Назва замовника	Форма зі списком угод
6	Пошук автомобілів заданої вантажопідйомності або вищої	Вантажопідйомність авто	Форма зі списком авто
7	Пошук інформації про водія та автомобіль за номером автомобіля	Номер автомобіля	Форма зі списком авто та водіїв
8	Пошук необроблених замовлень	Форма зі списком замовлень	Форма зі списком необроблених замовлень
9	Пошук вільних автомобілів	Форма зі списком авто	Форма зі списком вільних автомобілів
10	Пошук угоди за її номером	Форма зі списком угод	Форма з інформацією про обрану угоду
11	Пошук акту виконаних робіт та квитанції для обраної угоди	Номер угоди	Форма з інформацією про акт виконаних робіт та квитанцію
12	Розрахунок витрат, необхідних для виконання замовлення	Форма зі списком замовлень	Форма зі списком замовлень, та сумою витрат на виконання кожного замовлення
13	Формування звіту угод по кожному замовнику	Таблиці Замовник та Угода	Форма з сформованим звітом угод по кожному замовнику
14	Формування звіту замовлень по кожному замовнику	Таблиці Замовник та Замовлення	Форма з сформованим звітом замовлень по

			кожному замовнику
15	Формування звіту угод по кожному автомобілю	Таблиці Автомобіль та Угода	Форма з сформованим звітом угод по кожному автомобілю
16	Формування звіту угод за обраний місяць	Таблиця Угода та обраний місяць	Форма з сформованим звітом угод за обраний місяць

РОЗДІЛ 2. РОЗРОБКА СППР

2.1. Використані програмне забезпечення та технології

2.1.1. Використані програмні засоби

Для розробки даної системи були використані наступні програмні середовища та засоби:

- **AllFusion Erwin Process Modeler** — CASE-засіб для функціонального моделювання різних процесів з використанням нотацій IDEF0, IDEF3, DFD;
- **AllFusion Erwin Data Modeler** — CASE-засіб забезпечує дослідження предметної області з використанням об'єктно-орієнтованого підходу та проектування баз й сховищ даних для різних СУБД;
- **MS SQL Server 2008** — реляційна СУБД, яка виступає в ролі сервера для баз даних та може бути встановлено на різні види комп'ютерів (локально або серверно);
- **MS Visual Studio 2015** — інтегроване середовище фірми Майкрософт для проектування та створення інформаційних систем різної складності, з консольним чи графічним інтерфейсом, підтримує технологію Windows Form, а також веб-орієнтовані системи [5].

2.1.2. Використані технології

Для моделювання бізнес-процесу була використана нотація IDEF0, за допомогою якої були створені моделі “AS-IS” та “TO-BE”.

IDEF0 — методологія, що базується на графічному описі процесів на основі функціонального моделювання, що призначена на дослідження складних об'єктів автоматизації та бізнес-процесів.

IDEF0 підтримує ієрархічне представлення об'єктів, що дає змогу швидко проаналізувати предметну область. IDEF0 направлена на бізнес-

функції та логічні зв'язки між ними, а не на дотримання алгоритму їх виконання [10].

Для розробки інтерфейсу програмного засобу були використані програмна технологія Microsoft.NET та мова програмування C#.

Microsoft.NET — технологія та платформа для створення різних програмних додатків запропонована, запропонована фірмою Microsoft [9].

C# — об'єктно-орієнтована мова програмування з підтримкою платформи .NET [8].

Також для зміни дизайну програмного засобу була використана система керування пакетами NuGet.

NuGet — це вільна система керування пакунками, розроблена для Microsoft development platform[6].

Для виконання запитів до БД використовується мова запитів SQL.

SQL — мова запитів для роботи з об'єктами бази даних, що використовується для редагування, запису, видалення та вибірки даних. Крім того, вона забезпечує створення схеми бази даних та її модифікації, контролю доступом до елементів бази даних [7].

2.2. Проектування БД

Для проектування бази даних в першу чергу була розроблена логічна модель БД в програмному засобі AllFusion Erwin Data Modeler(додаток В). В логічній моделі БД представлені сутності, та їх взаємозв'язки в такому вигляді, в якому вони існують в реальному житті.

Після чого на основі логічної моделі БД була створена фізична модель БД, яка на відміну від логічної вже проектується з розрахунком того, під яку СУБД проектується БД, в нашому випадку це MS SQL Server 2008 (Додаток Г). На наступному кроці фізична модель БД була згенерована в БД в СУБД MS SQL Server 2008.

Для цього була попередньо створена порожня БД на сервері.

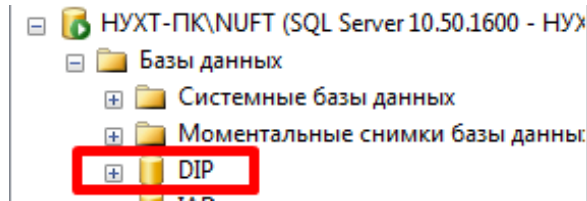


Рис. 2.1. Порожня БД

І обравши пункт меню “Schema generation” в меню Tools —> Forward Engineer в AllFusion Erwin Data Modeler згенерували схему в БД.

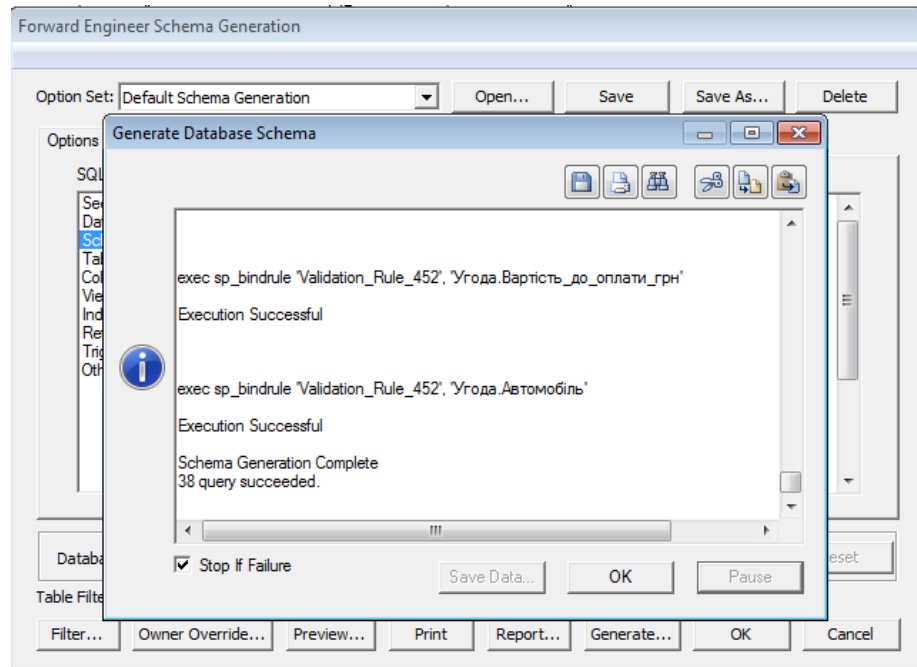


Рис.2.2. Генерація схеми БД

На цьому робота в AllFusion Erwin Data Modeler закінчується, і далі для подальшої розробки будуть використовуватися MS Visual Studio та MS SQL Server Management Studio.

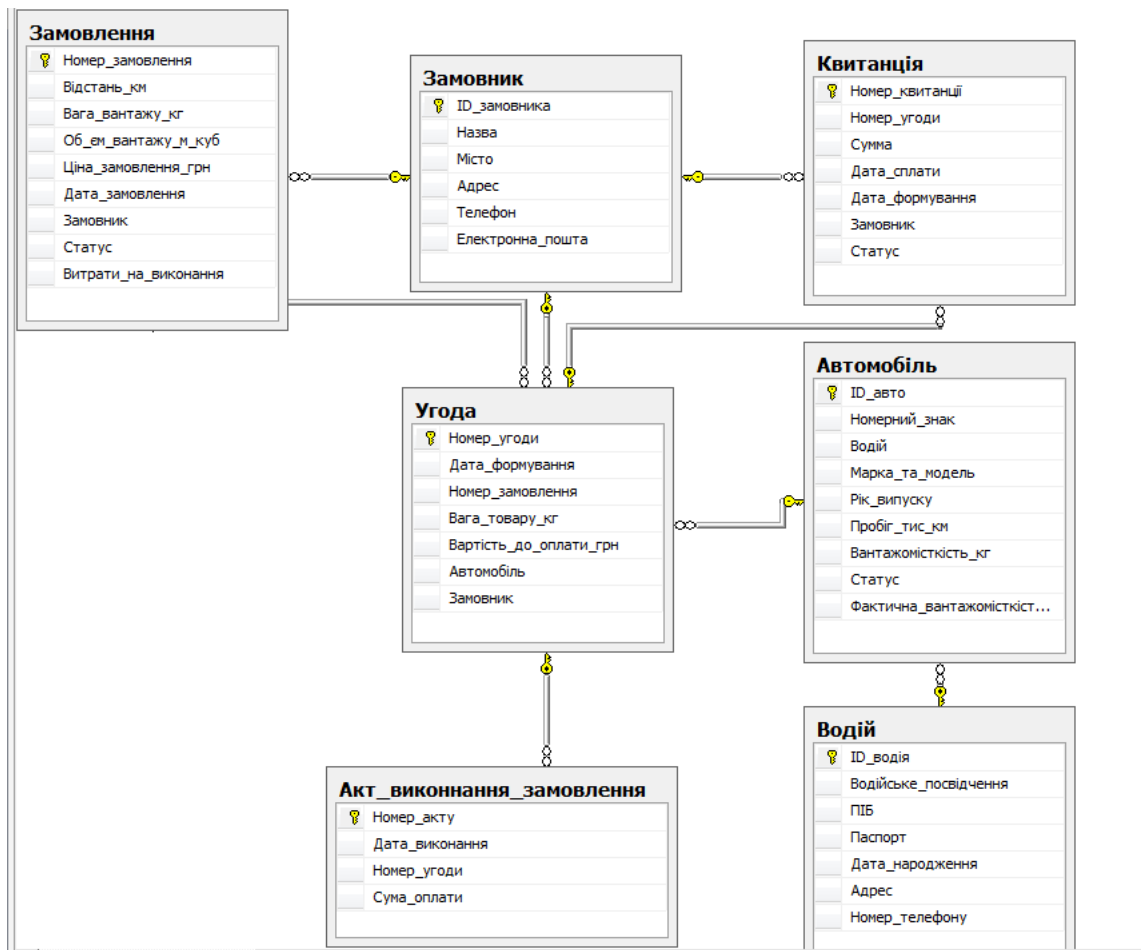


Рис. 2.3. Схема БД в MS SQL Server Management Studio

PK-первинний ключ таблиці, FK- вторинний ключ, тобто поля, значення яких будуть підставлятися з інших таблиць.

Опис всіх таблиць, переліку полів та їх типів даних можна побачити в таблиці 2.1.

Таблиця 2.1. Опис таблиць БД

Таблиця	Назва поля	Тип даних
Замовлення	Номер замовлення	Int(PK)
	Відстань км	Int
	Вага вантажу кг	Int
	Об'єм вантажу м куб	Float
	Ціна замовлення грн	Money
	Дата замовлення	Datetime
	Замовник	Int(FK)

	Статус	Varchar(10)
	Витрати на виконання	Money
Замовник	ID замовника	Int(PK)
	Назва	Varchar(35)
	Місто	Varchar(20)
	Адрес	Varchar(35)
	Телефон	Varchar(11)
	Електронна пошта	Varchar(35)
	Квитанція	Номер квитанції
Номер угоди		Int(FK)
Сума		Money
Дата сплати		Datetime
Дата формування		Datetime
Замовник		Int(FK)
Статус		Varchar(10)
Угода	Номер угоди	Int(PK)
	Дата формування	Datetime
	Номер замовлення	Int(FK)
	Вага товару кг	Int
	Вартість до оплати грн	Money
	Автомобіль	Int(FK)
	Замовник	Int(FK)
Автомобіль	ID авто	Int(PK)
	Номерний знак	Varchar(9)
	Водій	Int(FK)
	Марка та модель	Varchar(35)
	Рік випуску	Int
	Пробіг тис км	Int
	Вантажомісткість кг	Int

	Статус	Varchar(10)
	Фактична вантажомісткість	Int
Акт виконання замовлення	Номер акту	Int(PK)
	Дата виконання	Datetime
	Номер угоди	Int(FK)
	Сума сплати	Money
Водій	ID водія	Int(PK)
	Водійське посвідчення	Varchar(11)
	ПІБ	Varchar(35)
	Паспорт	Varchar(9)
	Дата народження	datetime
	Адрес	Varchar(35)
	Номер телефону	Varchar(11)

2.3. Розробка інтерфейсу користувача

Для створення інтерфейсу користувача був використаний додаток Windows Forms в середовищі розробки MS Visual Studio 2015.

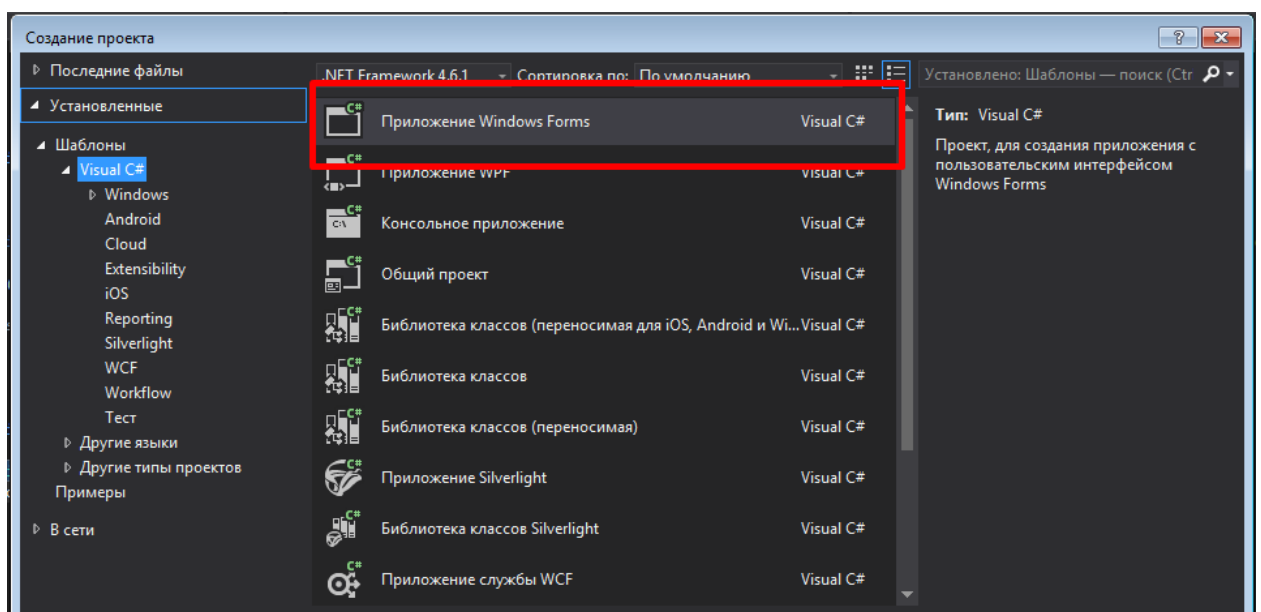


Рис.2.4. Додаток Windows Forms

При першому відкритті проекту в вікні Джерела даних буде відображатися наступне повідомлення.

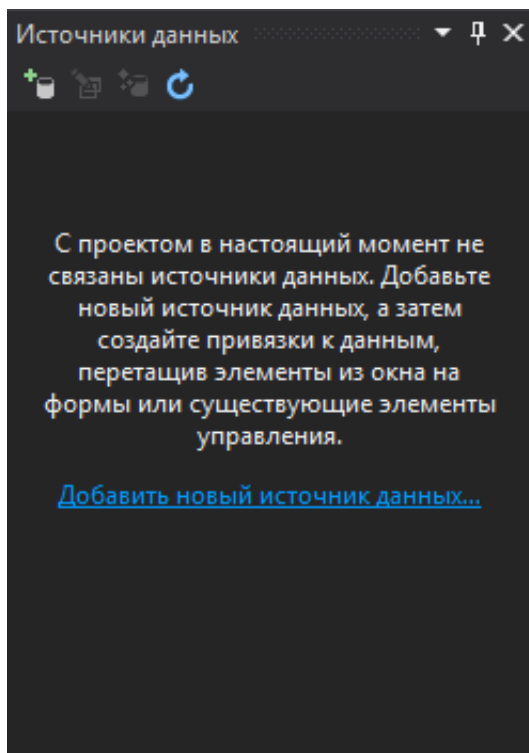


Рис.2.5. Джерела даних

Натиснувши на “Додати нове джерело даних....” та виконуючі інструкції створюємо зв’язок з БД.

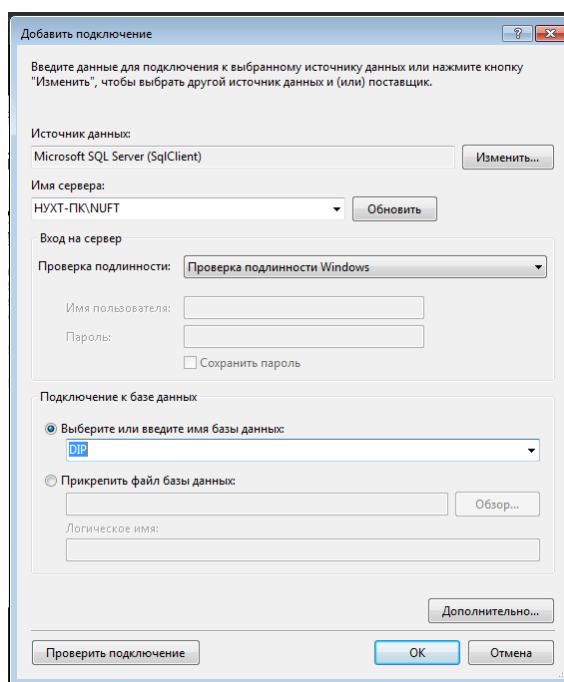


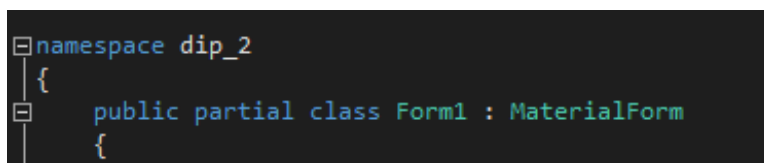
Рис.2.6. Додавання нового підключення

Після чого середовище автоматично сформувало рядок підключення проекту до БД. Він має наступний вигляд:

```
Data Source=HVXT-ПК\NUFT;Initial Catalog=DIP;Integrated Security=True
```

За допомогою контролю пакунків NuGet замість звичайної теми оформлення Windows Form, система буде мати тему оформлення Material Skin, яка змінить звичайний, нудний дизайн звичайних форм на більш сучасний, та привабливий.

Після того, коли в NuGet ми завантажили до нашого проекту пакет Material Skin, оберемо необхідну нам форму, перейдемо до її коду, та замінімо клас від якого унаслідується наша форма з Form на MaterialForm (Рис.2.7).



```
namespace dip_2
{
    public partial class Form1 : MaterialForm
    {
        ...
    }
}
```

Рис. 2.7. Зміна батьківського класу форми

На наступному етапі було створене головне вікно програми, та головне меню, за допомогою якого користувачі будуть переходити між вікнами додатка (Рис. 2.8.).

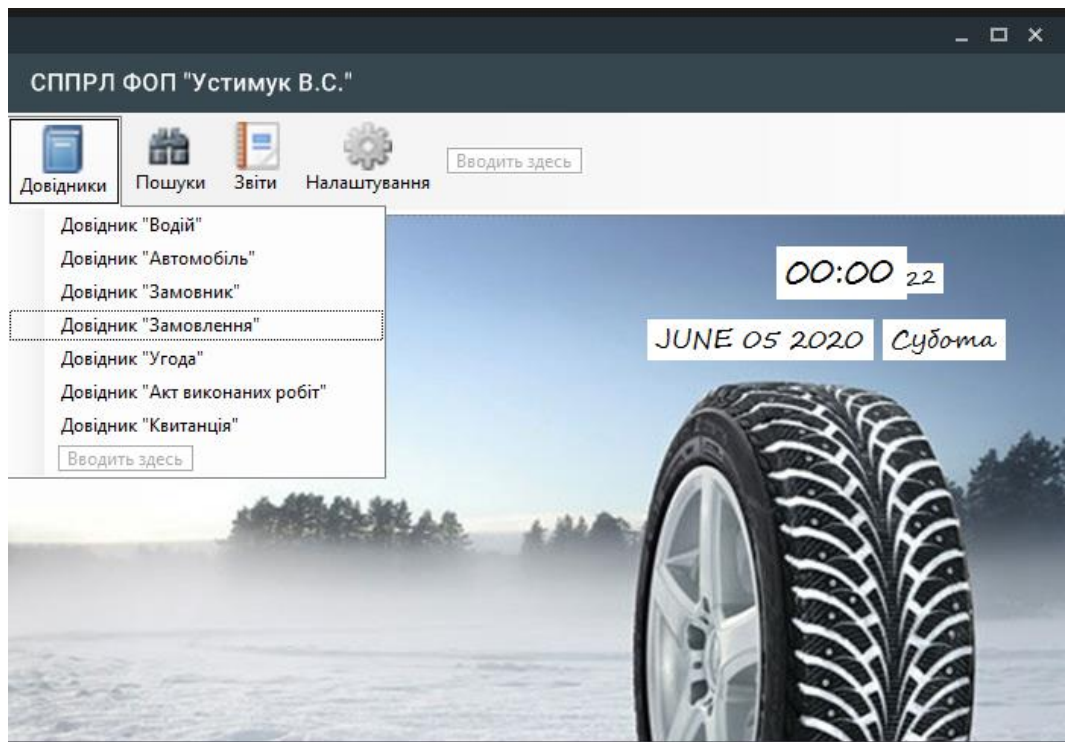


Рис. 2.8. Зовнішній вигляд головного вікна додатку

За допомогою меню, користувач може відкрити один з 7 довідників, скористатися одним з 5 пошуків або сформувавши один з 4 звітів. Також деякі з пошукових функцій системи реалізовані на сторінках довідників, детальніше це буде продемонстровано далі. Такий підхід зменшив к-сть пунктів меню, тим самим зменшив к-сть сторінок, необхідних для відвідування користувачу, покращив ергономічність програми, а також зменшив її розмір.

Кожен пункт меню відкриває нове вікно програми, з необхідним довідником, або пошуком (Рис. 2.9).

```
private void довідникВодійToolStripMenuItem_Click(object sender, EventArgs e)
{
    driver N = new driver();
    N.ShowDialog();
}
```

Рис. 2.9. Приклад програмного коду клавіші для відкриття нового вікна

Також для забезпечення захисту програмного засобу, при запускові програми в першу чергу користувачу необхідно виконати вхід в систему (Рис. 2.10.).

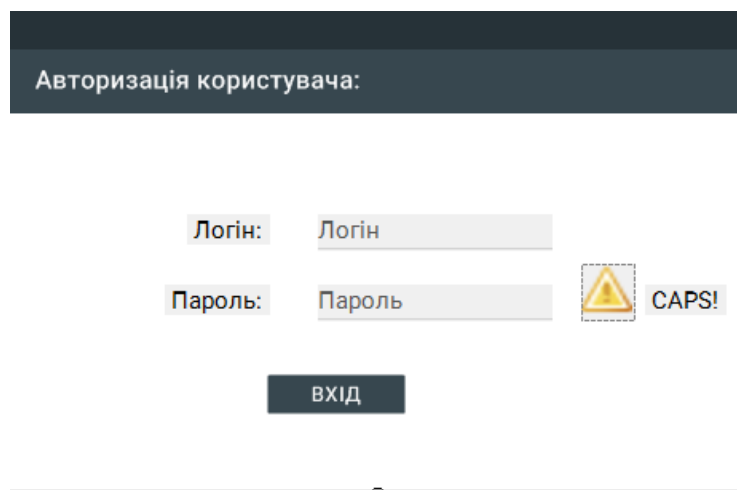


Рис. 2.10. Вікно авторизації

Як видно із Рис.2.10 поля для введення інформації містять в собі підказки, тому у користувачів не повинно виникнути проблем з входом до системи, а підказка з підписом “CAPS!” своїм появленням зверне увагу користувача на те, що можливо його невдача при спробі входу до системи пов’язана з увімкненою на його клавіатурі клавішею “CAPS-LOCK”.

Розглянемо як виглядає вікно довідника, а також його особливості та основні можливості на прикладі довідника “Автомобіль” (Рис. 2.11).

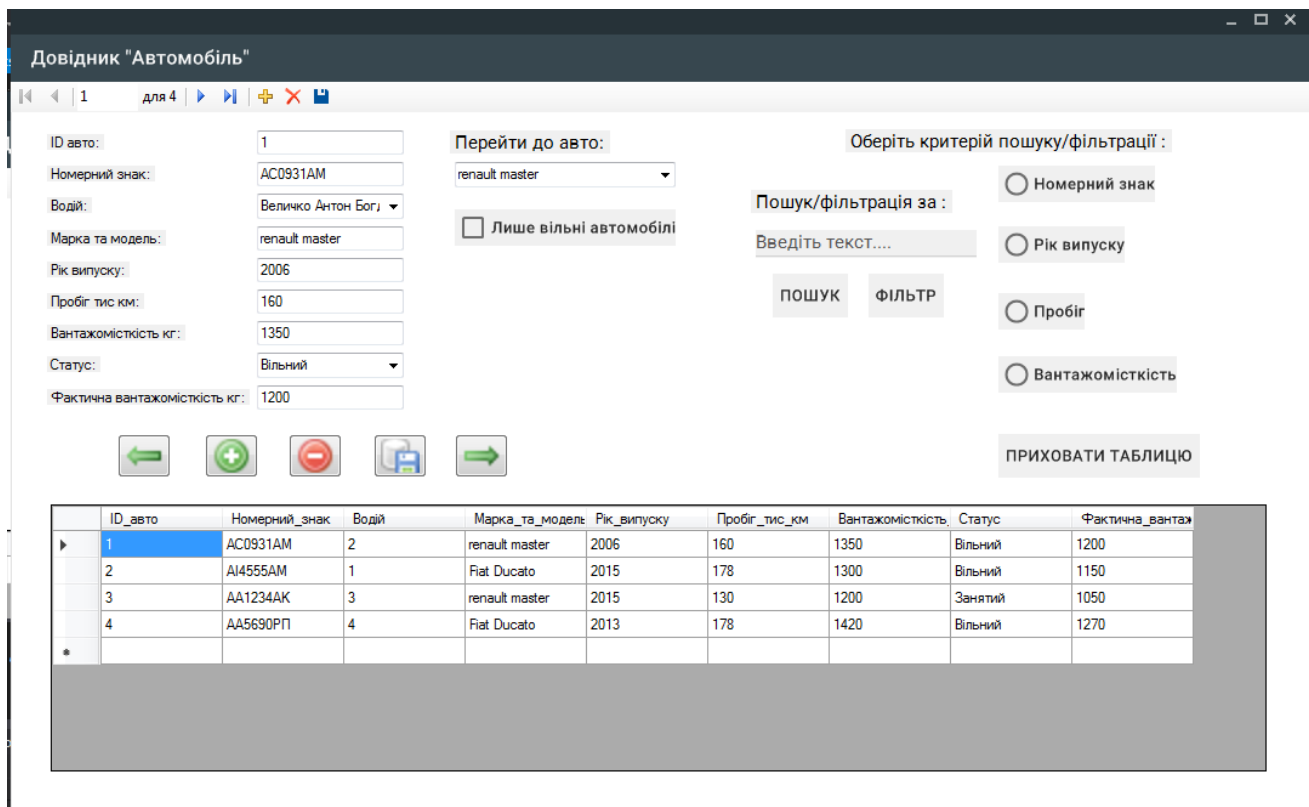


Рис. 2.11. Довідник "Автомобіль"

На рисунку 2.11 представлена форма по роботі з інформацією про автомобілі у вигляді таблиці та у вигляді опису кожного автомобіля по одному, таблиця по замовчуванню прихована, а кнопка "Приховати таблицю" має підпис "Показати таблицю". Програмний код, що забезпечує обробку по натисканню кнопки "Показати таблицю" представлено на рисунку 2.12. Це дає змогу користувачу переглянути таблицю, коли йому необхідно переглянути дані у стислому виді, проте працювати з даними звісно приємніше та зручніше коли кожне поле запису відображається в окремому текстовому полі.

```

private void materialFlatButton1_Click(object sender, EventArgs e)
{
    if (автомобільDataGridView.Visible == false)
    {
        автомобільDataGridView.Visible = true;
        materialFlatButton1.Text = "Приховати таблицю";
    }
    else
    {
        автомобільDataGridView.Visible = false;
        materialFlatButton1.Text = "Показати таблицю";
    }
}

```

Рис. 2.12. Код клавiшi для приховання/вiдображення таблицi

Також на формi дублюються кнопки, для переходу мiж записами, додавання, видалення записiв та збереження iнформацiї (Рис. 2.13).

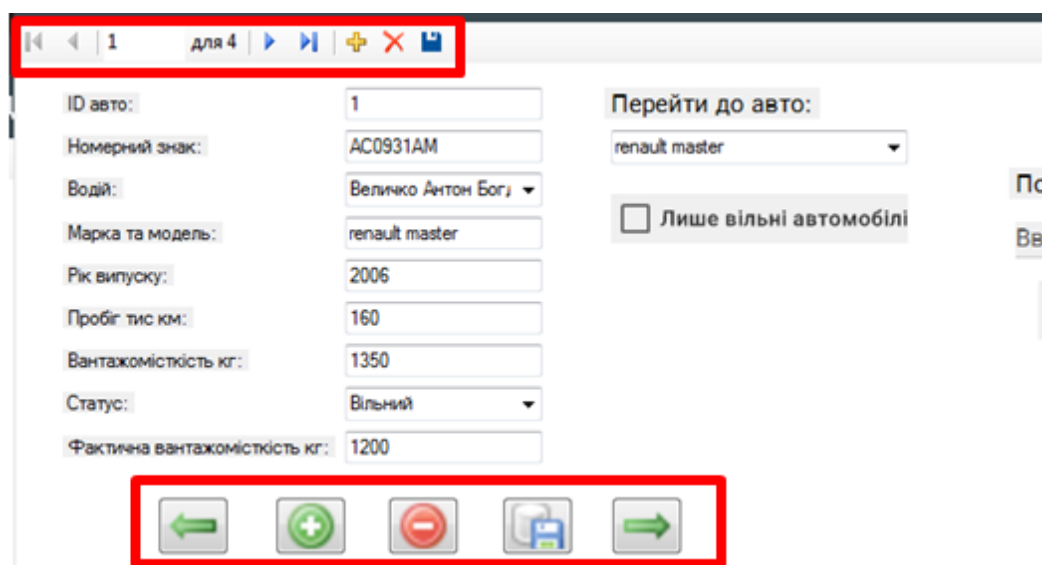


Рис. 2.13. Дублювання кнопок на формi

Це зроблено для зручностi користувача, оскiльки кнопки, зверху, якi автоматично створюються при додаванні на форму даних з Джерела даних дуже малi, та користуватися ними не завжди зручно.

iз зовнiшнього вигляду кнопок вiдразу зрозумiло яку функцiю має кожна з них.

В даному довіднику, деякі поля запису представлені в формі поля зі списком, це означає що ці поля обираються з інших таблиць, і за допомогою поля зі списком виводиться перелік можливих для вибору варіантів.

Для цього налаштування поля зі списком були змінені наступним чином.

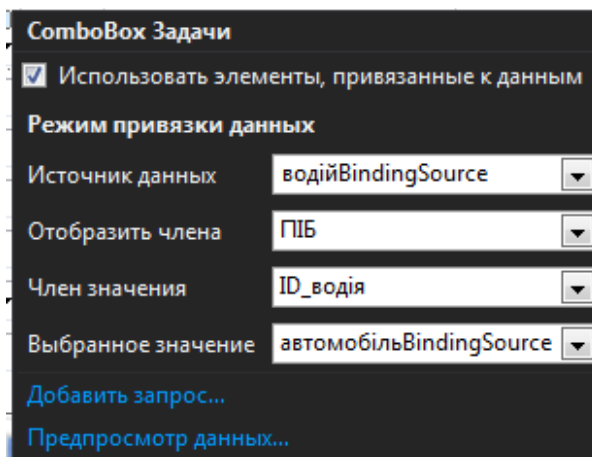


Рис. 2.14. Налаштування поля зі списком

За допомогою таких налаштувань, поле зі списком виводить всіх водіїв з таблиці “Водій”, а саме їх ПІБ, що дуже спрощує процес вибору водія.

Також на формі присутнє поле зі списком для швидкого переходу між автомобілями.

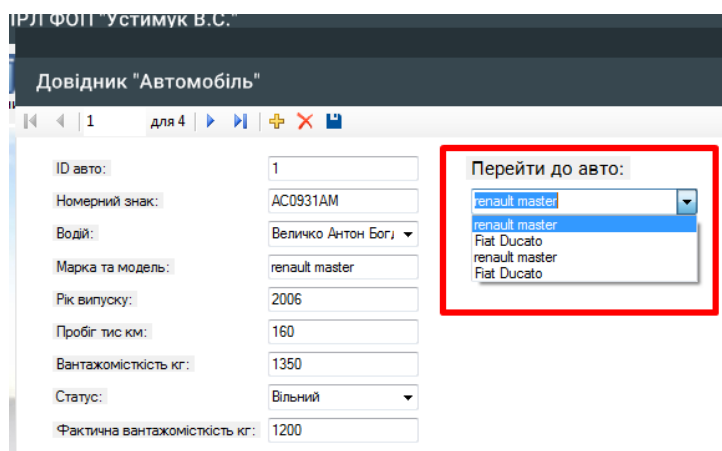


Рис. 2.15. Поле зі списком для швидкої навігації

На рисунку 2.15 відображені марка та модель всіх автомобілів фірми, а для переходу між записами досить просто обрати потрібний автомобіль зі списку.

За допомогою довідника можна додавати нові записи, редагувати вже існуючі або видаляти записи. Для реалізації всіх цих функцій непотрібно писати додаткові SQL-запити оскільки середовище розробки вже зробило це за нас, проте для зручності ми все ж таки внесли деякі зміни до цих запитів.

Фактична вантажомісткість — це поле, яке вираховується на основі вантажомісткості автомобіля, тому заповнення цього поля ми автоматизували, для цього ми перейшли в схему нашої БД в файлах проекту знайшли там tableadapter для таблиці “Автомобіль”, та в контекстному меню обрали пункт Налаштувати (Рис. 2.16).

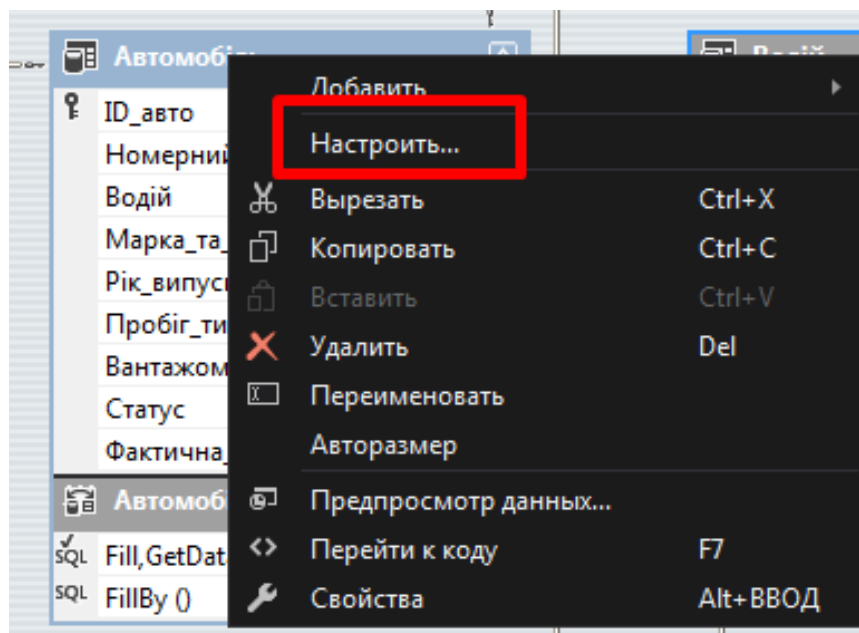


Рис. 2.16. Налаштування tableadapter'а

Змінили SQL-запит представлено на рисунку 2.17.

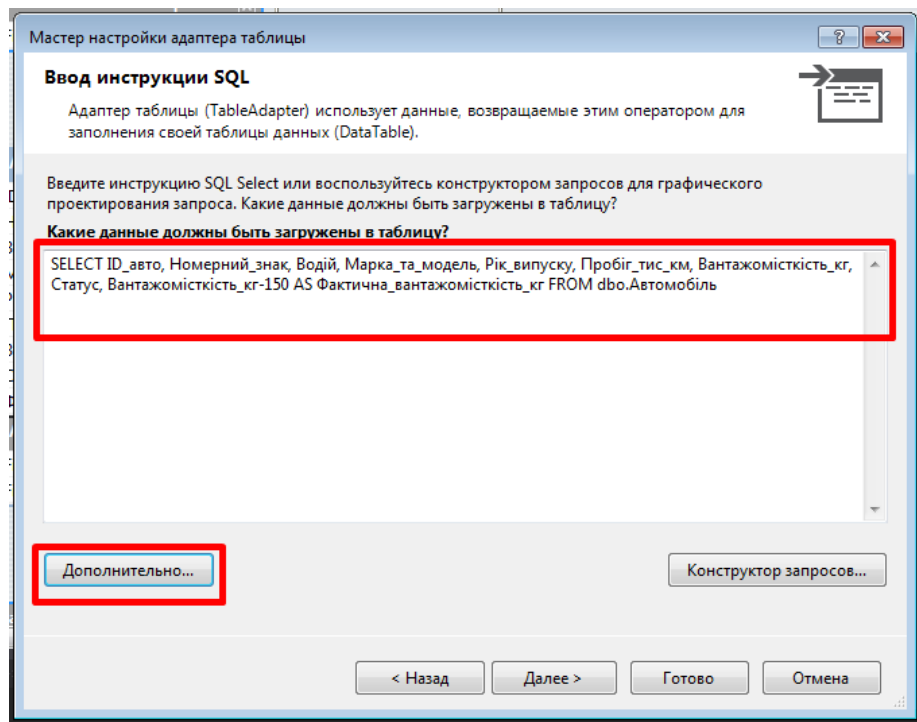


Рис. 2.17. Зміна SQL-запиту

Тепер, як бачимо з коду запиту, Фактична вантажомісткість буде вираховуватися за формулою “Вантажомісткість” – 150.

А якщо натиснути клавішу “Додатково”, то можна побачити, що проставлені необхідні галочки, які забезпечують автоматичне формування запитів на додавання до таблиці, оновлення та видалення, на основі Select-запиту, який зображений на рисунку 2.18

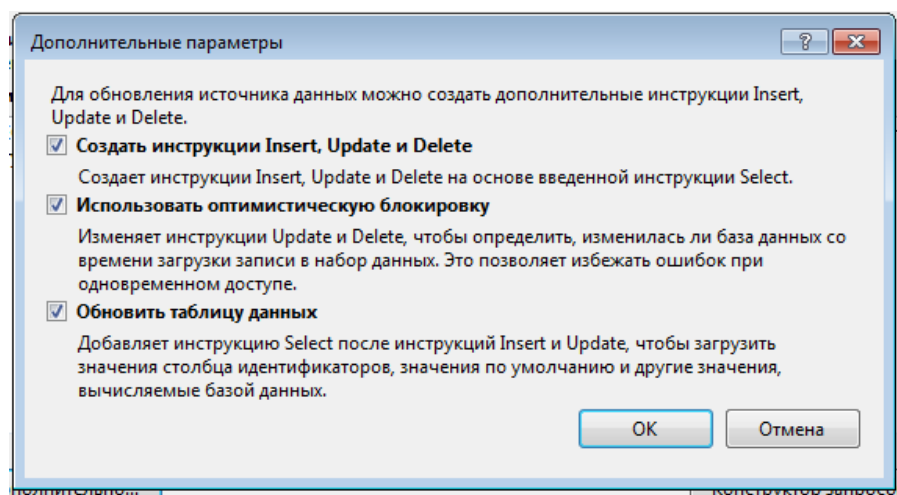


Рис. 2.18. Зміст кнопки “Додатково”

Аналогічним чином нами реалізована функція розрахунку вартості замовлення, це буде розглянуто нижче.

Ще одна функція, яка реалізована в довіднику це пошук вільних автомобілів, яка виглядає як прапорець “Лише вільні автомобілі” (Рис. 2.19).

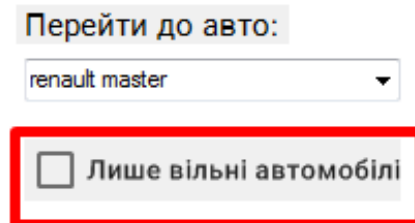


Рис. 2.19. Реалізація пошуку за допомогою прапорця

Активація цього прапорця, активує фільтр, за допомогою якого ми можемо відсіяти з переліку автомобілів всі, окрім тих, статус яких в даний момент “Вільний” (Рис. 2.20).

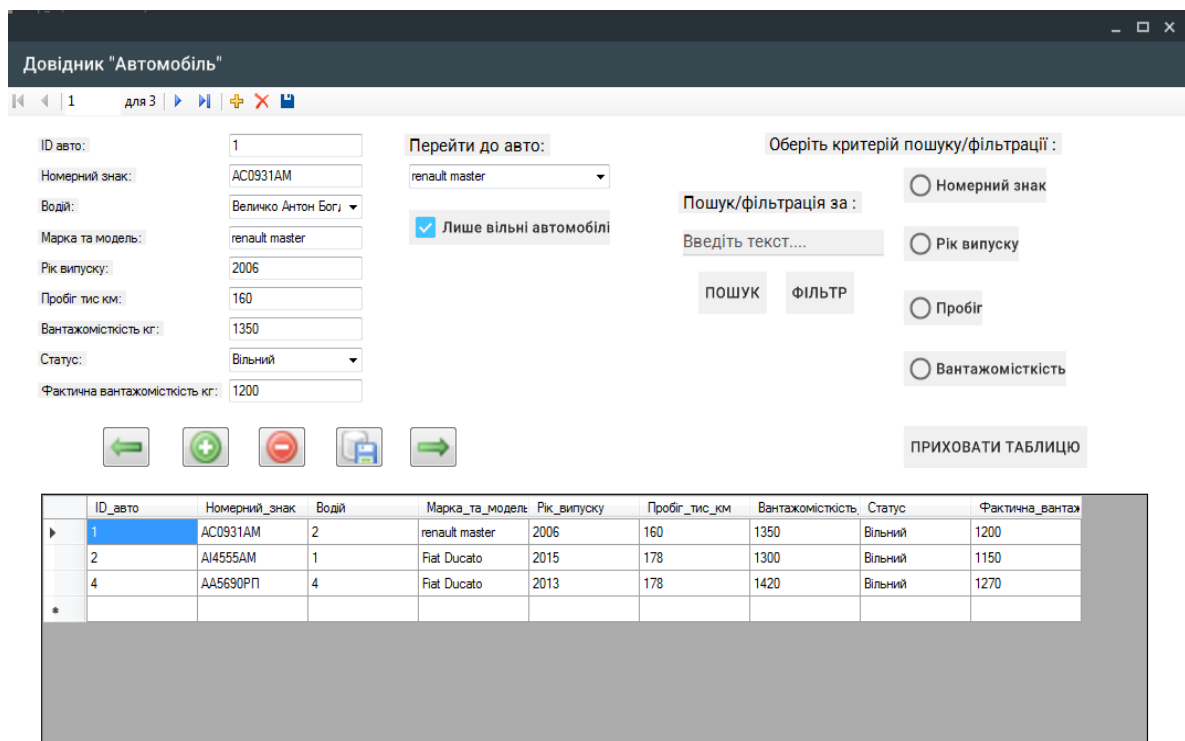


Рис. 2.20. Приклад роботи прапорця “Лише вільні автомобілі”

Програмна реалізація представлена на рисунку 2.21.

```
private void materialCheckBox1_CheckedChanged(object sender, EventArgs e)
{
    if (materialCheckBox1.Checked == true)
    {
        this.автомобільBindingSource.Filter = "Статус LIKE '%Вільн%'";
    }
    else
    {
        this.автомобільBindingSource.RemoveFilter();
    }
}
```

Рис.2.20. Код функції зміни стану прапорця

Також кожен довідник має свій критерій для фільтрації/пошуку, а в довіднику “Автомобіль” ми ще й можемо обрати критерій з переліку, це виглядає наступним чином.

Оберіть критерій пошуку/фільтрації :

Пошук/фільтрація за :

Введіть текст....

ПОШУК ФІЛЬТР

!!!Активованний фільтр!!!

Номерний знак

Рік випуску

Пробіг

Вантажомісткість

Рис.2.21. Реалізація пошуку/фільтрації з вибором критерія

Тобто користувач, може обрати один з 4 на вибір критеріїв за допомогою радіо-кнопок, після чого виконати пошук або фільтрацію.

Напис “активованний фільтр” по замовчуванню прихований, і відображається коли користувач активує фільтр, це зроблено для того, щоб неухважний користувач не панікував, коли раптом не зможе знайти записи, які тільки що були в таблиці, а зверне його увагу на те, що записи відсутні тільки через те, що він активував фільтр.

Коди кнопок Пошук та Фільтр представлені нижче на рисунках 2.22-2.23.

```

private void materialFlatButton2_Click(object sender, EventArgs e)
{
    int itemFound=0;
    string param="";
    if (materialRadioButton1.Checked == true)
        param = "Номерний_знак";
    if (materialRadioButton2.Checked == true)
    {
        param = "Рік_випуску";
    }
    if (materialRadioButton3.Checked == true)
        param = "Пробіг_тис_км";
    if (materialRadioButton4.Checked == true)
        param = "Вантажомісткість_кг";
    if(param=="")
        MessageBox.Show("Оберіть критерій для пошуку", "Помилка");
    else
    {
        itemFound = this.автомобільBindingSource.Find(param, materialSingleLineTextField1.Text);
        if (itemFound < 0)
            MessageBox.Show("Відсутні записи з вашим значенням критерія", "Помилка");
        else
            if(materialSingleLineTextField1.Text.Length==0)
                MessageBox.Show("Некоректно введені дані для пошуку", "Помилка");
            else
                this.автомобільBindingSource.Position = itemFound;
    }
}
}

```

Рис.2.22. Код клавiші “Пошук”

```

private void materialFlatButton3_Click(object sender, EventArgs e)
{
    if (materialLabel3.Visible == false)
    {
        int value=0;
        string param = "";
        if (materialRadioButton1.Checked == true)
        {
            param = "Номерний_знак";
            this.автомобільBindingSource.Filter = param + " LIKE '%'" + materialSingleLineTextField1.Text + "%'";
        }
        if (materialRadioButton2.Checked == true)
        {
            param = "Рік_випуску";
            int.TryParse(materialSingleLineTextField1.Text, out value);
            this.автомобільBindingSource.Filter = param + " = " + value;
        }
        if (materialRadioButton3.Checked == true)
        {
            param = "Пробіг_тис_км";
            int.TryParse(materialSingleLineTextField1.Text, out value);
            this.автомобільBindingSource.Filter = param + " = " + value;
        }
        if (materialRadioButton4.Checked == true)
        {
            param = "Вантажомісткість_кг";
            int.TryParse(materialSingleLineTextField1.Text, out value);
            this.автомобільBindingSource.Filter = param + " = " + value;
        }
        if (param == "")
            MessageBox.Show("Оберіть критерій для фільтрації", "Помилка");
        else
        {
            materialLabel3.BackColor = Color.Red;
            materialLabel3.Visible = true;
        }
    }
    else
    {
        this.автомобільBindingSource.RemoveFilter();
        materialLabel3.Visible = false;
    }
}
}

```

Рис.2.23. Код клавiші “Фільтрація”

Як можна помітити, також в кодi клавiш присутні блоки, для захисту від некоректних дій користувача, а саме пошук та фільтрація, не будуть виконуватися, доки користувачем не буде обраний критерій або не будуть введені необхідні дані.

При роботі з програмою можна чітко прослідкувати певні алгоритми в дії користувача. Наприклад, коли на підприємстві з'являється новий автомобіль, також буде і новий водій, тоді зрозуміло, що після додавання водія, користувач буде додавати новий автомобіль, саме тому на формі довідника "Водій" є кнопка "довідник "Автомобіль", для швидкого переходу між довідниками.

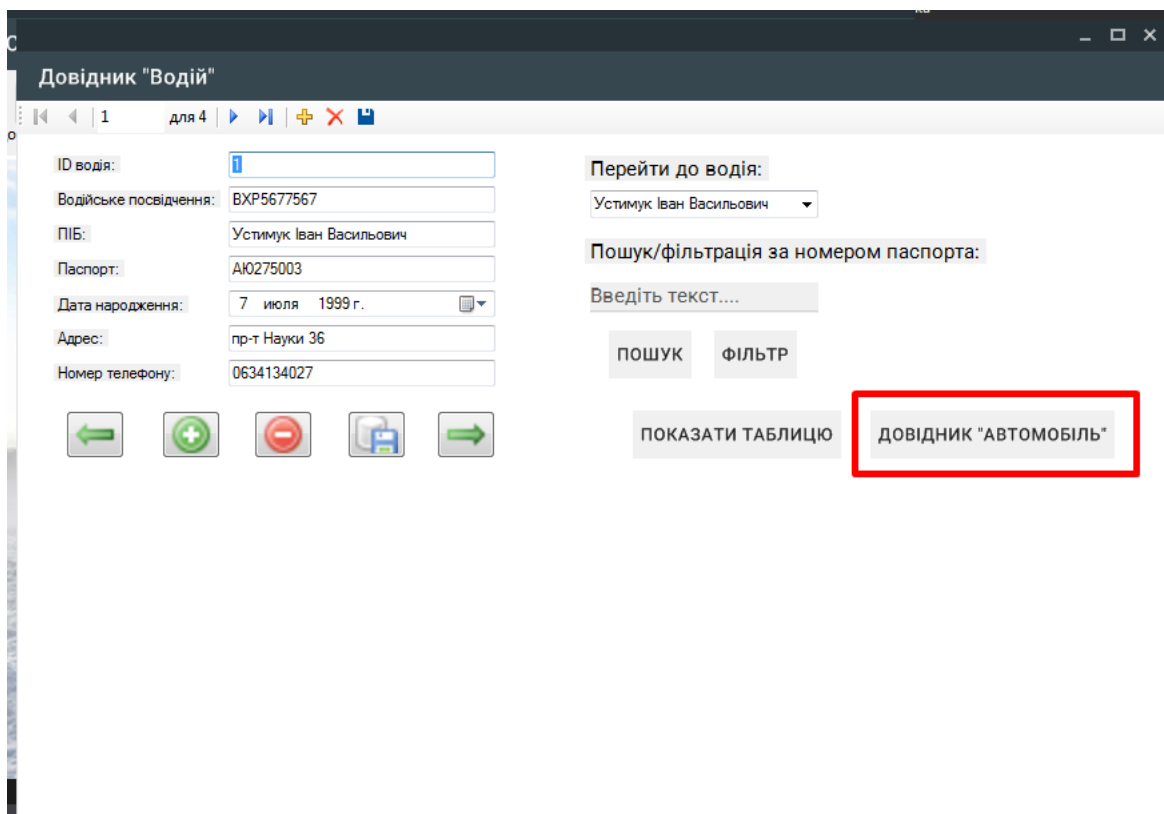


Рис.2.24. Клавiша для швидкого переходу між довідниками

У всьому іншому, всі довідники мають спільну структуру. Функція розрахунку вартості замовлення здійснено повністю автоматична, і реалізована таким же чином як розрахунок фактичної вантажомісткості автомобіля.

Для цього SQL-запит tableadaptera Замовлення, модифікований наступним чином:

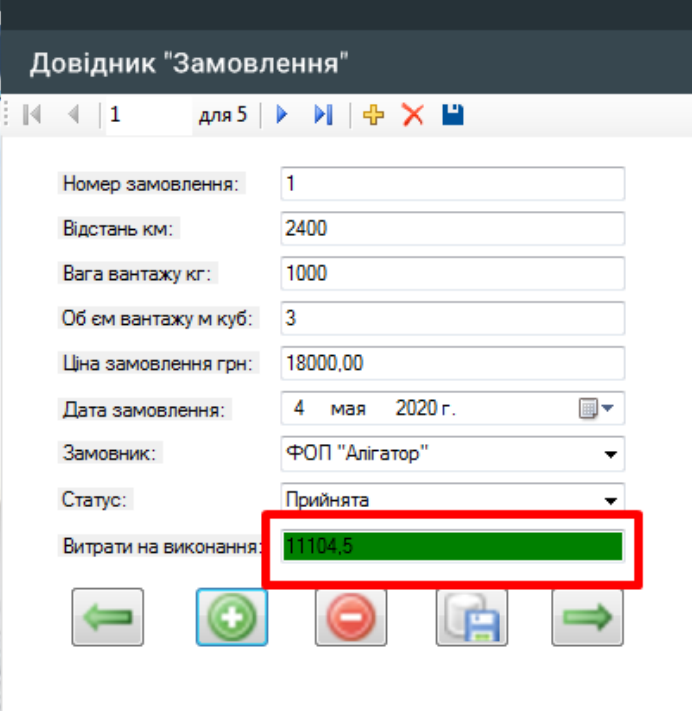
```
SELECT   Номер_замовлення,   Відстань_км,   Вага_вантажу_кг,  
Об_єм_вантажу_м_куб,   Ціна_замовлення_грн,   Дата_замовлення,   Замовник,
```

Статус,

$(Об_єм_вантаж_м_куб*1.5+Вага_вантаж_кг*1.5)+(Відстань_км*4) AS$
Витрати на виконання **FROM** dbo.Замовлення

Після введення всіх даних про замовлення, досить просто натиснути клавішу зберегти, і система автоматично обрахує вартість замовлення за формулою в якій присутній об'єм, вага вантажу та відстань, яку необхідно подолати. А сам довідник "Замовлення" має механізм сигналізації про економічну вигідність даного замовлення.

Це реалізовано зміною кольору текстового поля, в якому відображається розмір витрат. Поле стає зеленим, коли різниця між витратами та ціною більше 5 тисяч гривень(мінімальна різниця для отримання прибутку), або червоним, коли різниця менше 5 тисяч гривень.



The screenshot shows a software interface titled "Довідник 'Замовлення'". It contains several input fields for order details: "Номер замовлення" (1), "Відстань км" (2400), "Вага вантажу кг" (1000), "Об'єм вантажу м куб" (3), "Ціна замовлення грн" (18000,00), "Дата замовлення" (4 мая 2020 г.), "Замовник" (ФОП "Алігатор"), and "Статус" (Прийнята). The field "Витрати на виконання" (11104,5) is highlighted in green and is enclosed in a red rectangular box. Below the form are several navigation buttons: a left arrow, a green plus sign, a red minus sign, a document icon, and a right arrow.

Рис. 2.25. Демонстрація зміни кольору поля в залежності від вигідності замовлення

Поле	Значення	Символ
Номер замовлення:	3	П
Відстань км:	3000	З
Вага вантажу кг:	670	П
Об'єм вантажу м куб:	1	Ві
Ціна замовлення грн:	16000.00	
Дата замовлення:	4 мая 2020 г.	
Замовник:	ФОП "Алігатор"	
Статус:	Прийнята	
Витрати на виконання:	13006.5	

Рис. 2.26. Демонстрація зміни кольору поля в залежності від вигідності замовлення

Програмна реалізація представлена на рисунку 2.27.

```
private void витрати_на_виконанняTextBox_TextChanged(object sender, EventArgs e)
{
    витрати_на_виконанняTextBox.BackColor = Color.White;
    decimal value=0,value1;
    decimal.TryParse(витрати_на_виконанняTextBox.Text, out value);
    value = Math.Round(value, 2);
    витрати_на_виконанняTextBox.Text = value.ToString();
    if (value>0)
    {
        decimal.TryParse(ціна_замовлення_грнTextBox.Text, out value1);
        if (value >= value1 - 5000)
            витрати_на_виконанняTextBox.BackColor = Color.Red;
        else
            витрати_на_виконанняTextBox.BackColor = Color.Green;
    }
}
```

Рис.2.27. Програмний код зміни кольору поля в залежності від вигідності замовлення

Для реалізації пошуків на формах, нам необхідно було створити додаткові tableadapter, розмістити їх на формах, та запрограмувати кнопки для пошуку.

Розглянемо реалізацію пошуку на прикладі пошуку “Замовлення замовника”.

В першу чергу був створений додатковий tableadapter “замовлення_замовника”, а його SELECT-запит виглядає наступним чином:

```
SELECT    Замовлення.Номер_замовлення, Замовлення.Відстань_км,  
Замовлення.Вага_вантажу_кг, Замовлення.Об_єм_вантажу_м_куб,  
Замовлення.Ціна_замовлення_грн,  
Замовлення.Дата_замовлення, Замовлення.Замовник, Замовлення.Статус,  
Замовлення.Витрати_на_виконання  
  
FROM    Замовлення INNER JOIN  
Замовник    ON    Замовлення.Замовник    =  
Замовник.ID_замовника  
  
WHERE    (Замовник.Назва = @p1)
```

Саме останній рядок SQL-запиту, робить цей запит параметричним, що означає для його виконання, нам необхідно буде ввести певну інформацію. Це ми зможе побачити коли розглянемо форму, яка була створена на наступному етапі.

Пошук "Замовлення замовника"

Номер замовлення:

Відстань км:

Вага вантажу кг:

Об'єм вантажу м куб:

Ціна замовлення грн:

Дата замовлення: 1 июня 2020 г.

Замовник:

Статус:

Витрати на виконання:

Оберіть замовника

ФОП "Алігатор"

ФОП "Алігатор"

ПП "Корнелюк А.А."

ТОВ "Омега"

ЗНАЙТИ ЗАМОВЛЕННЯ

ПОКАЗАТИ ТАБЛИЦЮ

Рис.2.28. Приклад форми для пошуку

Вигляд форми для пошуку не дуже відрізняється від форми довідника, про те деякі зміни все ж таки присутні, відсутні елементи для фільтрації

даних, а також кнопки додавання, видалення та збереження, адже вони тут непотрібні.

Також, відсутня можливість редагування даних, адже це пошук, а не довідник.

Так як цей пошук використовується для пошуку замовлень обраного замовлення, то ми позбавили користувача від необхідності введення назви замовника вручну, і замінили це поле на поле зі списком, з переліком всіх замовників із таблиці “Замовник”, про цей механізм вже не раз йшлося в даній роботі.

Деякі кнопки вибули, проте є й одна, якої у нас немає на формах довідників, це кнопка для друку сторінки (Рис .2.29).

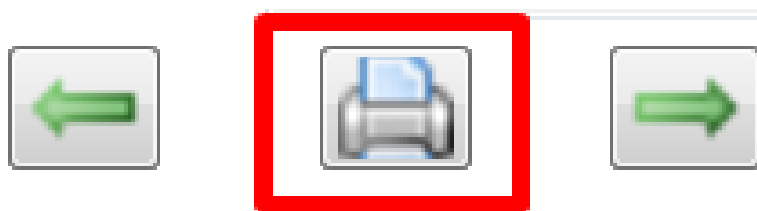


Рис.2.29. Клавiша для друку результатiв пошуку

А програмна реалiзацiя представлена на рисунку 2.30.

```
private void button2_Click(object sender, EventArgs e)
{
    PrintDialog dlg = new PrintDialog();
    dlg.ShowDialog();
}
```

Рис.2.30. Код клавiшi для друку

А в результатi вiдкривається вiкно Windows для друку (Рис. 2.31).

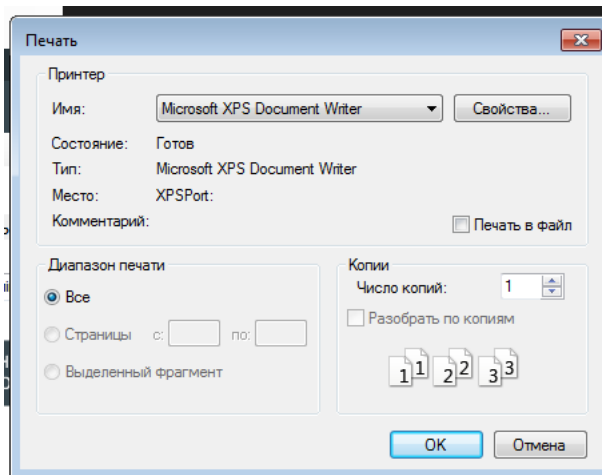


Рис.2.31. Вікно друку

І останній механізм, який був використаний, це механізм формування звітів. Розглянемо його на прикладі звіту “Угоди за обраний місяць”.

Для його реалізації в першу чергу був створений та налаштований новий елемент — майстер звітів. Його можна знайти на вкладці Reporting.

Також для реалізації цього звіту нам знадобився новий tableadapter, оскільки для цього звіту нам необхідно обрати місяць, за яким формуватися звіт.

SQL-запит даного tableadapter-а має наступний вигляд.

```

SELECT                                Замовник.Назва, Угода.Номер_угоди,
Угода.Дата_формування,                Угода.Вага_товару_кг,
Угода.Вартість_до_оплати_грн

FROM      Угода INNER JOIN
              Замовник ON Угода.Замовник = Замовник.ID_замовника

WHERE      (YEAR(Угода.Дата_формування) = @p1) AND
              (MONTH(Угода.Дата_формування) = @p2)

```

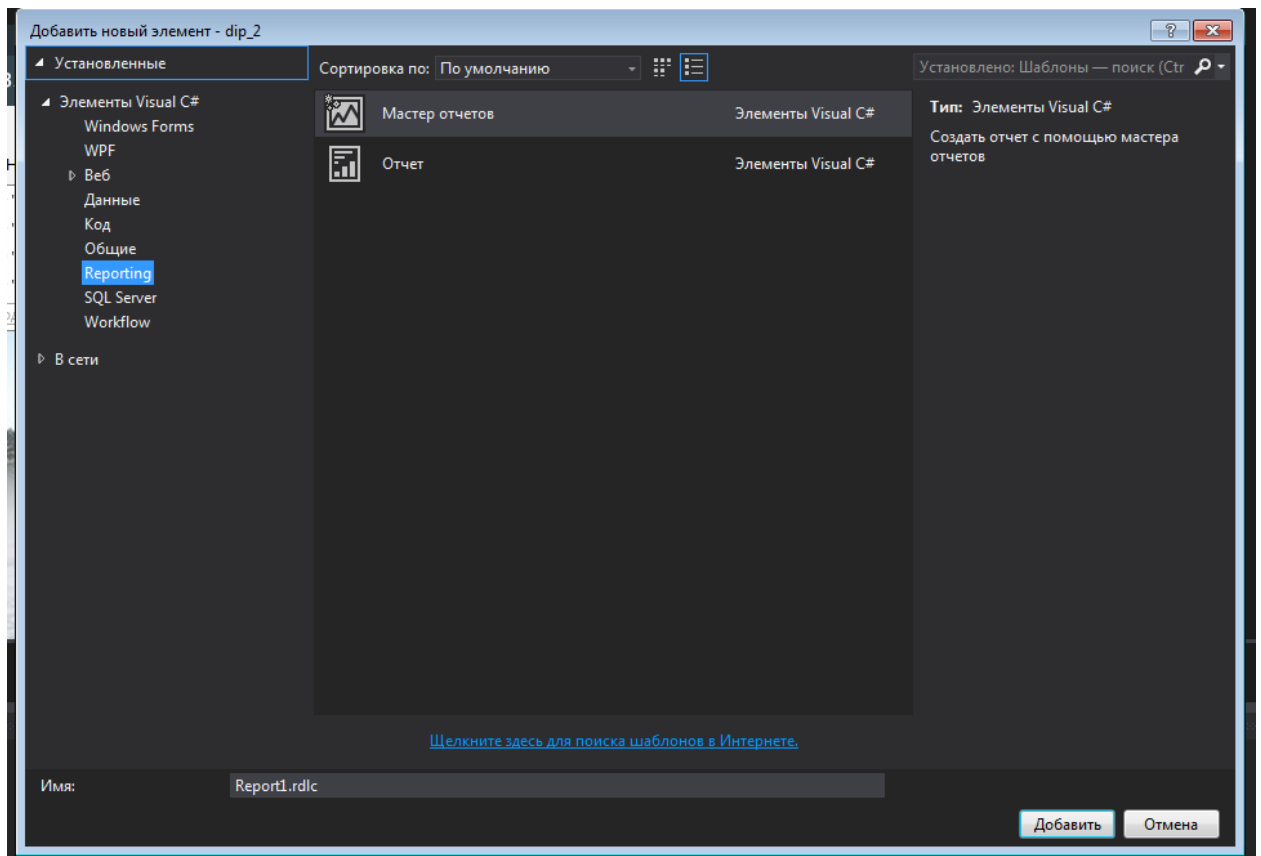


Рис.2.32. Элемент “Майстер звітів”

Після чого була створена нова форма, до неї доданий елемент report viewer. А в налаштуваннях даного елемента був обраний відповідний звіт.

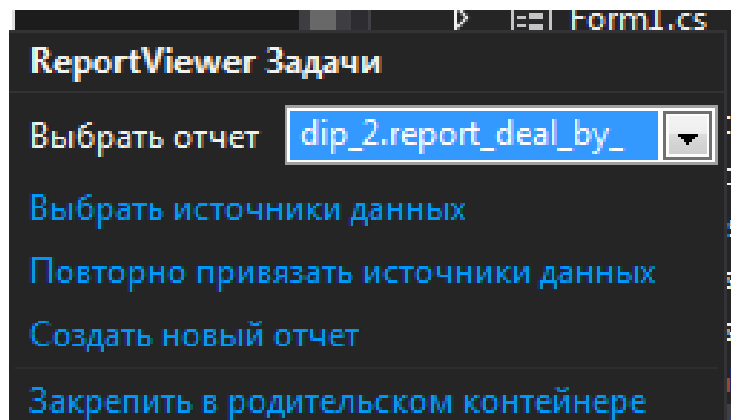


Рис.2.33. Налаштування елемента report viewer

Форма має вигляд, що представлено на рисунку 2.34, а програмний код представлено на рисунку 2.35.

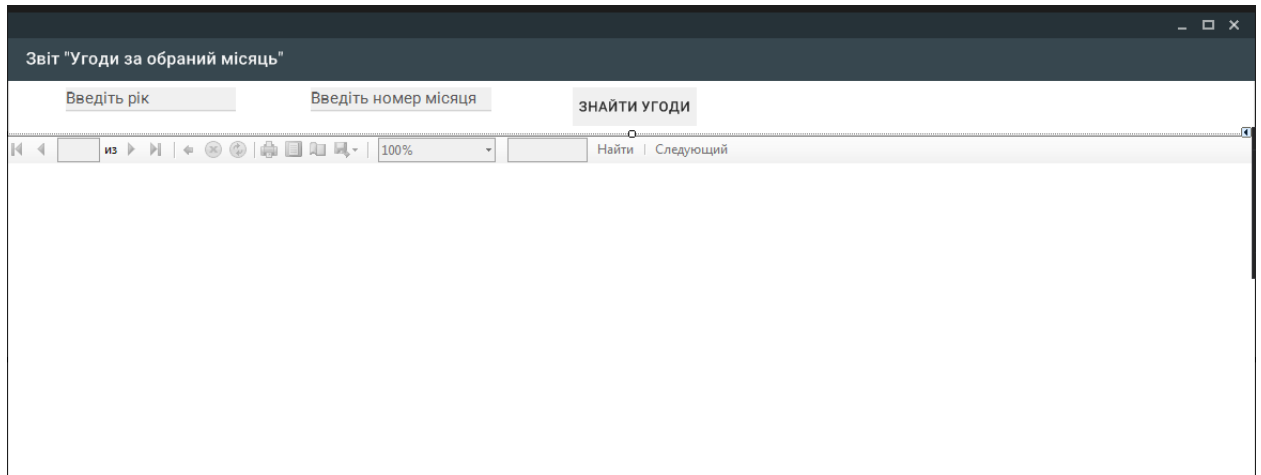


Рис.2.34. Приклад форми для звіту

```
private void materialFlatButton1_Click(object sender, EventArgs e)
{
    try
    {
        this.deals_by_monthTableAdapter.Fill(this.dIPDataSet.deals_by_month, ((decimal)(System.Convert.ChangeType(materialSingleLineTextField1.Text,
        typeof(decimal))))), ((decimal)(System.Convert.ChangeType(materialSingleLineTextField2.Text, typeof(decimal))));
        this.reportViewer1.RefreshReport();
        if(reportViewer1.Visible != true)
            reportViewer1.Visible = true;
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}
```

Рис.2.35. Код клавiші для формування звіту

Поля для введення даних містять в собі підказки, куди і яку інформацію необхідно вводити.

А всі інші звіти формуються після натиску на кнопки на вкладці меню “Звіти”, оскільки вони не потребують ніяких вхідних параметрів.

2.4. Інструкція користувача

Після встановлення на ПК співробітника підприємства даного програмного забезпечення, рекомендується перед використанням прочитати інструкцію користування кожному користувачу.

2.4.1. Вхід до системи

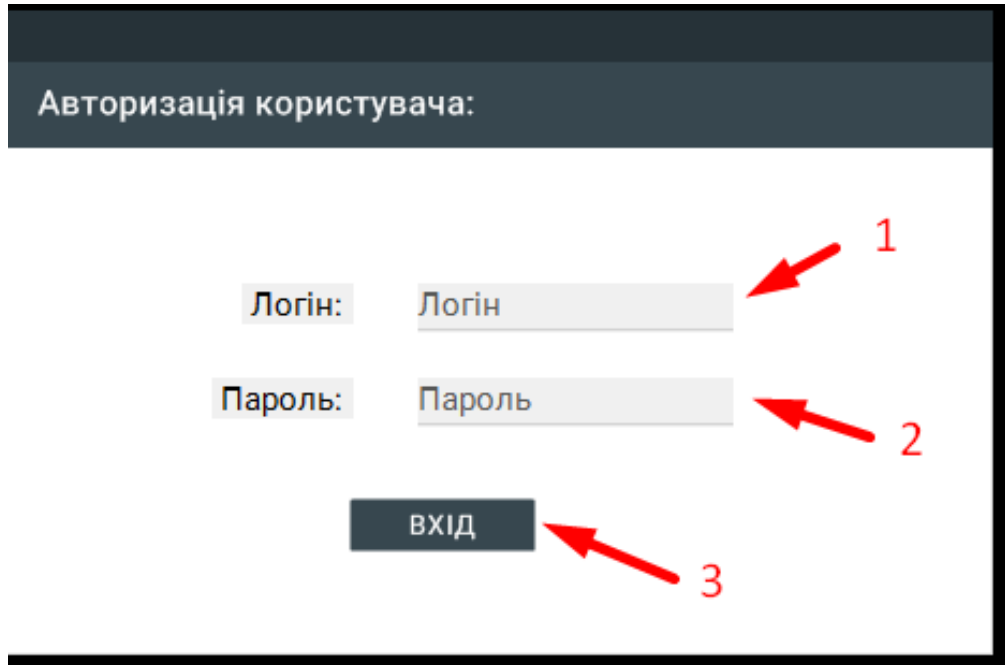


Рис. 2.36. Вікно авторизації

1. Поле для введення логіну
2. Поле для введення пароля
3. Кнопка для входу до системи

Використовуйте тільки свій логін та пароль, виданий адміністратором для входу до системи, та не забувайте, що “пароль” та “ПаРоЛЬ” це різні паролі.

Якщо після натискання клавіші “Вхід” додаткове вікно сповістило, що ваші дані введені не вірно, а на вікні входу з’явився напис “CAPS!”. Можливо ваш вхід був не успішним саме через те, що на вашій клавіатурі активована клавіша Caps-lock.

2.4.2. Головне вікно програми

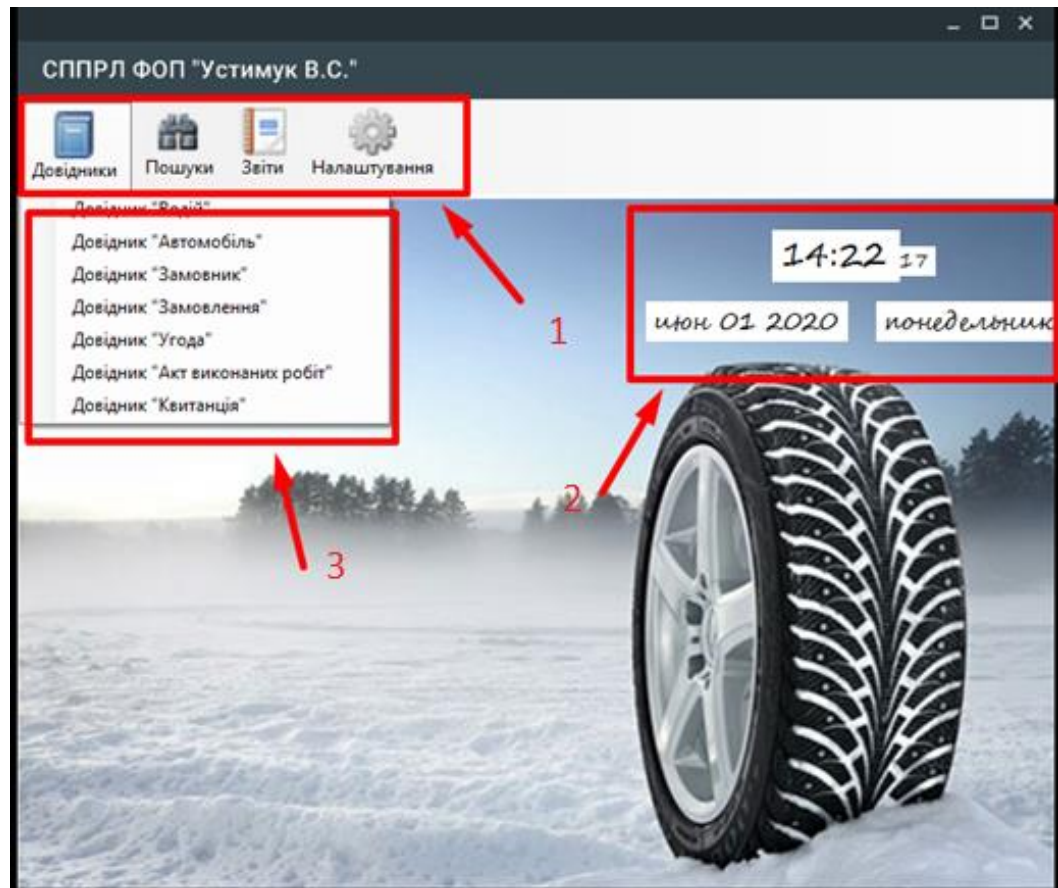


Рис. 2.37. Головне вікно програми

1. Головне меню програми, а при натискові на одну з його пунктів, буде відображений перелік форм, які можна відкрити.
2. Поточний час та дата.
3. Випадаюче меню, яке відкриється після натиску на кнопку головного меню “Довідники”.

За допомогою головного меню ви можете переміщатися між функціоналом програми, який вам необхідний. А пункт меню “Налаштування” надасть вам змогу вийти з програми, або покинути свій обліковий запис.

2.4.3. Робота з довідниками

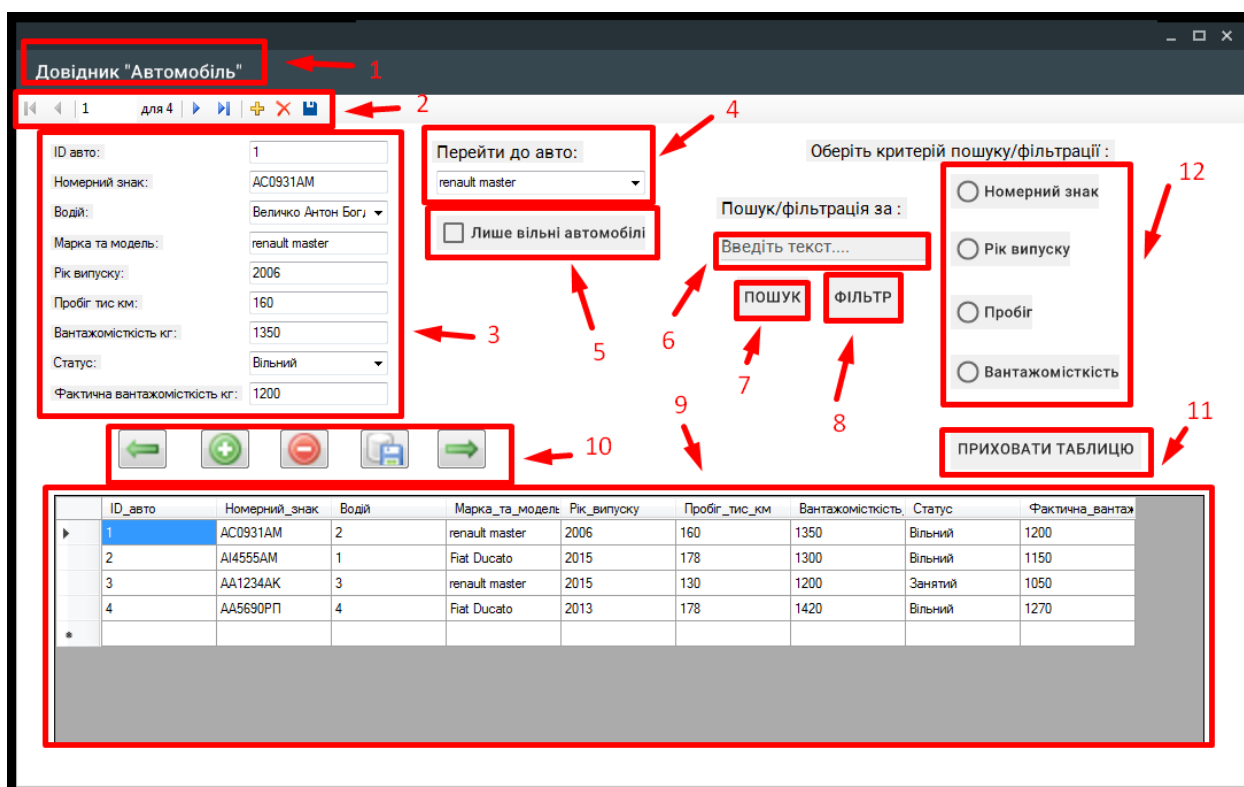


Рис. 2.38. Вікно довідника “Автомобіль”

1. Назва довідника
2. Клавiші для перемiщення мiж записами, додавання нового запису, видалення запису або збереження даних.

УВАГА!!! Не забувайте натискати клавiшу “Зберегти” пiсля редагування, видалення або додавання запису.

3. Представлення запису таблиці в деталізованій формі
4. Поле зі списком для переміщення між записами
Для переміщення між записами вам досить просто обрати необхідний вам запис з поля зі списком.

5. Прапорець “Лише вільні автомобілі”
Дозволяє вам відфільтрувати список автомобілів, та залишити лише ті, які на даний момент вільні. Зверніть увагу, чи не активований прапорець, коли вам здається, що не всі записи відображаються.

6. Поле для введення значення критерія для пошуку або фільтрації

7. Клавіша для активації пошуку за обраним вами критерієм

Не забувайте перед натиском клавіші обрати критерій (пункт №12 на рисунку 2.38) та ввести дані в поле №6.

8. Клавіша для активації фільтру за обраним вами критерієм

Не забувайте перед натиском клавіші обрати критерій (№12 на рисунку 2.38) та ввести дані в поле №6.

А також зверніть увагу, якщо на формі відображається напис “Активований фільтр”, це означає що зараз фільтр активний, і для того щоб його зняти, необхідно ще раз натиснути на клавішу “Фільтр”

9. Табличне представлення даних

По замовчуванню воно приховане, але його можна активувати натиском на кнопку №11.

10. Збільшене представлення клавіш №2 для більш зручної роботи з ними

11. Кнопка для активації/приховування таблиці №9

12. Критерії для пошуку/фільтрації

Не забудьте обрати один з критеріїв до того як натискати клавіші №7 або №8.

2.4.4. Робота з пошуками

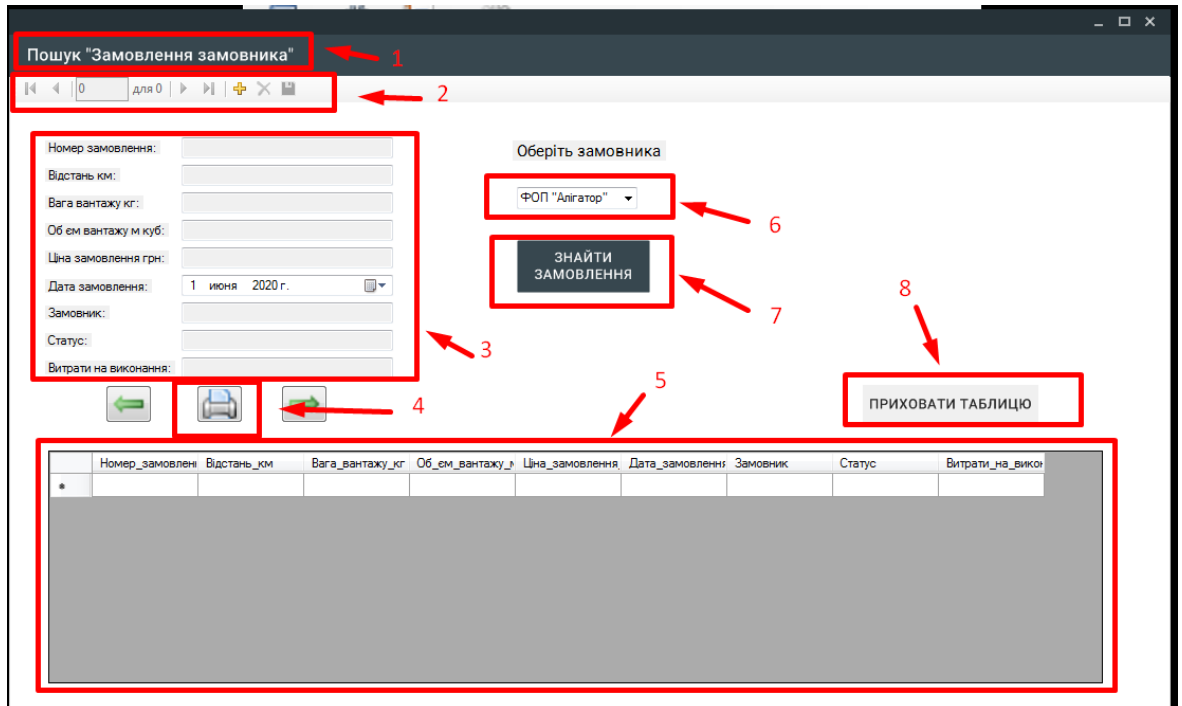


Рис. 2.39. Вікно пошуку “Замовлення замовника”

1. Назва пошуку
 2. Клавіші для переміщення між записами, додавання нового запису, видалення запису або збереження даних.
 3. Представлення запису таблиці в деталізованій формі
 4. Кнопка для відкриття вікна друку сторінки
 5. Табличне зображення даних
 6. Поле зі списком для вибору критерія пошуку
- Не забувайте обрати параметр для пошуку з поля зі списком перед тим як натискати клавішу №7.
7. Клавіша для виконання пошуку
 8. Кнопка для активації/приховування таблиці №5

УВАГА!!! Не забувайте що пошук, це не довідник, редагувати, додавати або видаляти дані тут заборонено. З цією метою поля заблоковані для змін.

2.4.5. Робота зі звітами

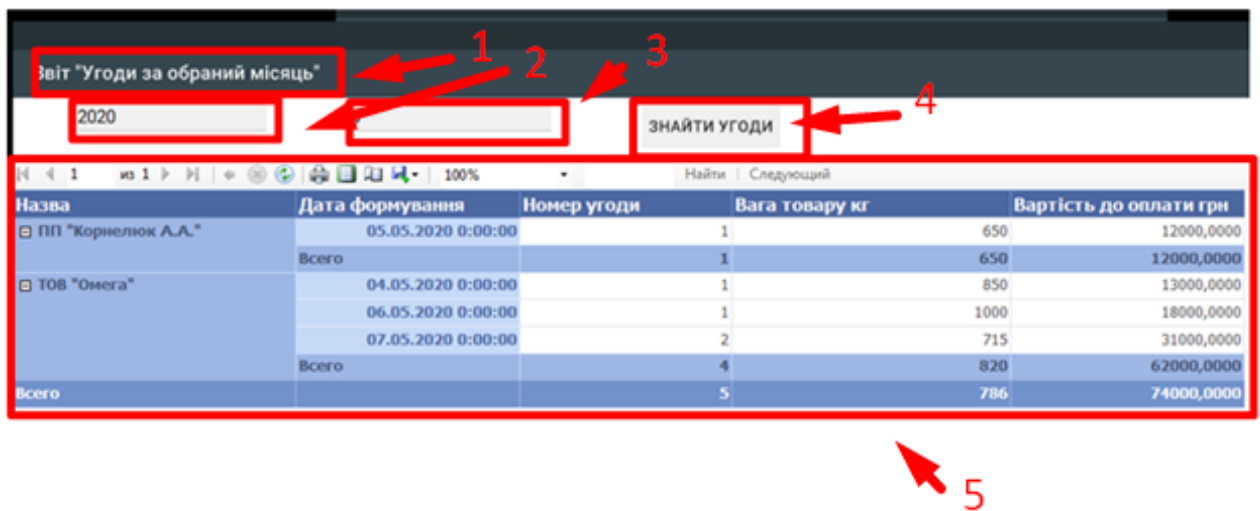


Рис. 2.40. Вікно звіту “Угоди за обраний місяць”

1. Назва звіту
2. Поле для введення року
3. Поле для введення місяця

Поле №2 та №3 не мають підписів, проте коли вони пусті, на них відображаються підказки.

Введіть рік

Введіть номер місяця

Рис.2.41. Вигляд пустих полів

4. Кнопка для формування звіту за обраним місяцем та роком
5. Вигляд сформованого після натиску кнопки №4 звіту

Зверніть увагу, що у верхній частині звіту №5 розміщені всі необхідні клавіші для роботи зі звітом, в тому числі для його друку.

2.5. Техніко-економічне обґрунтування

2.5.1. Розрахунок оплати праці

Вид системи: управління транспортними перевезеннями.

Ступень новизни розроблюваних задач - В.

Група складності алгоритму: Алгоритми обліку, звітності, статистики пошуку.

Визначення виду інформації представлена у таблиці 2.2.

Таблиця 2.2. Визначення виду інформації.

Вид інформації	Позначення	Кількість наборів даних
Кількість видів змінної інформації	ЗІ	$m=6$
Обробка в режимі реального часу	РЧ	Так
Кількість банків (баз) даних	БД	$p=1$
Кількість видів нормативно-довідкової інформації	НДІ	$n=2$
Забезпечення телекомунікаційної обробки даних і УО	ТОУ	Ні

Визначення необхідного часу на створення ескізного проекту T_1 та T_3
 T_2 :

Ескізний проект, $T_1 = 70$

Технічне завдання, $T_2 = 43$.

Визначення необхідного часу на етапі «технічний проект», «робочий проект» і «впровадження»:

- кількість форм вхідної інформації $B_1 = 3$;
- кількість форм вихідної інформації $B_2 = 3$;
- базове значення витрат часу для стадій "Технічний проект": $TБЗ = 19$;
- базове значення витрат часу для стадій "Робочий проект": $TБЗ = 96$;
- базове значення витрат часу для стадій "Впровадження": $TБЗ = 25$.

Табл. 2.3. Коефіцієнти k_1, k_2, k_3 для стадії "Технічний проект"

Вид використаної інформації	Ступінь новизни
-----------------------------	-----------------

	В
К1 (ЗІ)	1.0
К2 (НДІ)	0.72
К3 (БД)	2.08

Табл. 2.4. Коефіцієнти k_1, k_2, k_3 для стадії "Робочий проект"

Вид використаної інформації	Ступінь новизни
	В
К1 (ЗІ)	1.1
К2 (НДІ)	0.58
К3 (БД)	0.48

Табл. 2.5. Коефіцієнт ступеню новизни проекту, $k(o)$

Стадія розробки системи	Вид оброки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

$$K(c) = 1.00$$

Визначення витрат часу для стадії «технічний проект» (ТЗ)

$$k_{\pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}$$

$$K(\pi) = (1*6+0.72*1+2.08*1)/(6+2+1) = 0.97$$

$$T3 = K(\pi) * K(o)$$

$$T3 = 19 * 0.97 * 1.26 = 23.22$$

Визначення витрат часу на стадії «робочий проект» (Т4)

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}$$

$$K(\Pi) = (1.1*6+0.58*2+0.48*1)/(6+2+1) = 0.849$$

$$T4 = ТБЗ * K(\Pi) * K(o) * K(c)$$

$$T4 = 96 * 0.849 * 1.32 * 1 = 107.59$$

Визначення витрат часу на стадії «впровадження» (Т5)

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}$$

$$K(\Pi) = 0.849$$

$$T5 = ТБ5 * K(\Pi) * K(o) * K(c)$$

$$T5 = 25 * 0.849 * 1.21 * 1 = 21.23$$

Визначення загальних витрат часу на розробку системи:

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5$$

$$= 70 + 43 + 23.22 + 107.59 + 21.23 = 266 \text{ (годин)}$$

2.5.2. Визначення чисельності виконавців

Визначення чисельності виконавців вираховується за наступною формулою:

$$Ч = \frac{T_{\Sigma}}{\Phi}$$

Для дипломного проекту кількість робочих годин складає 500 із 7-годинним робочим днем. Враховуючі все, приймаємо на розробку проекту Φ , днів:

$$\Phi = 500/7 = 72.$$

Для дипломного проекту $\Phi = 72$ днів. Враховуючі 25 робочих днів в місяці.

Кількість місяців на розробку, М:

$$M = \Phi/25 = 72/25 = 3.$$

Для виконання цього проекту чисельність виконавців обраховується за:

$$\Phi = T(\text{загальне})/\Phi = 266/72 = 3.455 = 3$$

Оплата праці виконавців підраховується за формулою:

$$V_1' = \Phi * M * З_{ПР} = 3 * 3 * 16\,000 = 144\,000 \text{ грн}$$

2.5.3. Витрати, пов'язані з розробкою інформаційної системи:

- розрахунок річного фонду часу роботи ПК в годинах:

$$T_{ПК} = 3200 - (6*8+5*12) = 3092 \text{ год}$$

$$T'_{ПК} = 3092*(440/3200) = 425 \text{ год}$$

- поточні витрати на експлуатацію V_1'' :

$$Ц_{ПК} = Ц_p * (1 + k_{уН})$$

$$Ц_{ПК} = 10\,000 * (1+0.12) = 12\,200 \text{ грн.}$$

$$З_{АМ} = \frac{Ц_{ПК}}{H_A}$$

$$З_{АМ} = 12\,200/5 = 2\,440 \text{ грн}$$

$$З_{ЕЛ} = P_{ПК} * T_{ПК} * Ц_{ЕЛ} * A = 214.02 \text{ грн}$$

$$P_{ПК} = 0.3 \text{ кВт,}$$

$$T_{ПК} = 425 \text{ год,}$$

$$Ц_{ЕЛ} = 2.1 \text{ грн/кВт,}$$

$$A = 0.8.$$

$$З_p = Ц_{ПК} * 0.06$$

$$З_p = 12\,200 * 0.06 = 732 \text{ грн}$$

$$З_{МАТ} = Ц_{ПК} * 0.05$$

$$З_{МАТ} = 12\,200 * 0.05 = 610 \text{ грн}$$

$$V_1'' = З_{ОП} + З_{АМ} + З_{ЕЛ} + З_p + З_{МАТ}$$

$$V''_1 = 6\,000 + 2\,440 + 214.02 + 732 + 610 = 9996$$

$$V_1 = V'_1 + V''_1$$

$$V_1 = 144\,000 + 9996 = 153\,996 \text{ грн.}$$

- витрати на придбання і встановлення ПК V2. Так як на підприємстві кожен співробітник офісу вже має ПК, то:

$$V_2 = 0$$

- витрати на підготовку приміщення V3. У нас вже є приміщення підприємства базі якого і буде проходити розробка інформаційної системи, тому:

$$V_3 = 0$$

- витрати на підготовку та вивчення персоналу V4. Більшість співробітників офісу якщо і не досягли рівня «впевненого користувача» ПК, то на дуже близькому до його рівні, тому особливого навчання персонал не потребує, хоча провести декілька невеликих майстер-класів варто. Для цього фахівець має затратити свій час, тому:

$$V_4 = 2000 \text{ грн}$$

2.5.4. Загальна вартість розробки і впровадження системи

Загальна вартість системи (що враховує розробку і впровадження) вираховується:

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4$$

$$V = 153996 + 0 + 0 + 2000 = 155996 \text{ грн}$$

$$V_p = \frac{V_{\Sigma}}{H_A}$$

$$V(p) = 155996/5 = 31\,199,2 \text{ грн}$$

Таблиця 2.6. Основні джерела прибутку від впровадження інформаційної системи і порядок його підрахунку

№	Джерела прибутку	Порядок підрахунку прибутку	Сума за рік, грн
1.	Економія часу на зведення підсумкових даних	За рахунок автоматичного створення звітної документації	10 000
2.	Зменшення часових затрат на обробку нових замовлень	Всі дані містяться в одній системі, а певні критерії вираховуються автоматично	20 000
3.	Зменшення витрат на друк документів	За рахунок впровадження електронного обліку документів	20 000
Разом			50 000

$$K_{\text{ЕФ}} = \frac{\Pi_p}{V_p}$$

$$K_{\text{ЕФ}} = 50000/31\ 199,2 = 1.6$$

$$T_{\text{ок}} = 1/1.6 = 0.6$$

Таким чином, ми можемо сказати, що орієнтовний термін окупності системи буде становити приблизно 7 місяців.

РОЗДІЛ 3. ОХОРОНА ПРАЦІ

3.1. Аналіз умов праці

При роботі з розроблюваною системою, користувачі будуть проводити багато часу за комп'ютером, а це означає що користувачі можуть піддаватися впливу наступних негативних чинників: вплив електромагнітних полів, недостатнє освітлення, недостатня ергономіка робочого місця.

Цей негативний вплив може привести до погіршення стану здоров'я працівників, різних захворювань, що негативно вплине на роботу підприємства.

Для уникнення наведених вище проблем, або принаймні їхньої мінімізації, необхідно правильно організувати робочі місця персоналу та роботу персоналу в цілому.

3.2. Характеристика та організація робочого місця

Організація робочих місць проводиться згідно з ГОСТ 12.2.032-78.

Монітор має розміри з підставкою: 389.5 x 374.5 x 165.5 мм., розмір екрана по діагоналі – 43,18 см. Потрібно звернути увагу, що при збільшенні розміру екрану виникають певні недоліки: погіршується відбивна здатність екрану, тобто утворюється більше блисків; деформуються знаки на периферії екрана; виникає складність при розміщенні екрану у полі зору працівника.

Плоский екран, який поглинає зовнішні світлові потоки, з ергономічної точки зору є найкращим, адже він поглинає світлові потоки.

Оптимальною висотою розташування екрана є $50-35^\circ$ відносно до горизонталі, що відповідає напрямку зору користувача персонально комп'ютера. Якщо верхній край екрана вищий за рівень очей, то зчитування інформації з екрана ПК може викликати стан дискомфорту.

Робочі місця з персональним комп'ютером відповідають таким вимогам [11]:

- відстань між бічними поверхнями монітора – 1,2 м;

- відстань від тильної поверхні одною монітора до екрана іншого монітора – 2,5;
- прохід між рядами робочих місць – 1 м;
- висота робочої поверхні робочого стола – 680-800 мм;
- ширина робочої поверхні робочого стола 600-1400 мм;
- глибина робочої поверхні робочого стола – 800-1000 мм;
- простір для ніг; заввишки – 600мм. завширшки – 500мм, завглибшки – 450 мм.

Конструкція робочого місця користувача персонального комп'ютера має забезпечує норми для оптимальної робочої пози оператора за наступними ергономічними характеристиками [11]:

- ступні ніг розташовуються на підставці або на підлозі;
- стегна розташовані в горизонтальній площині;
- передпліччя розташовані вертикально;
- лікті розташовані під кутом 70-90 градусів відповідно вертикальної площини;
- зап'ястя зігнуті до 20 градусів до горизонтальної площини;
- нахил голови – 15-20 градусів відносно вертикальної площини.

Екран монітора та клавіатура розташовані не ближче 60 см від очей користувача, але з урахуванням розміру елементів (знаків і символів) на екрані, а також його розмір по діагоналі.

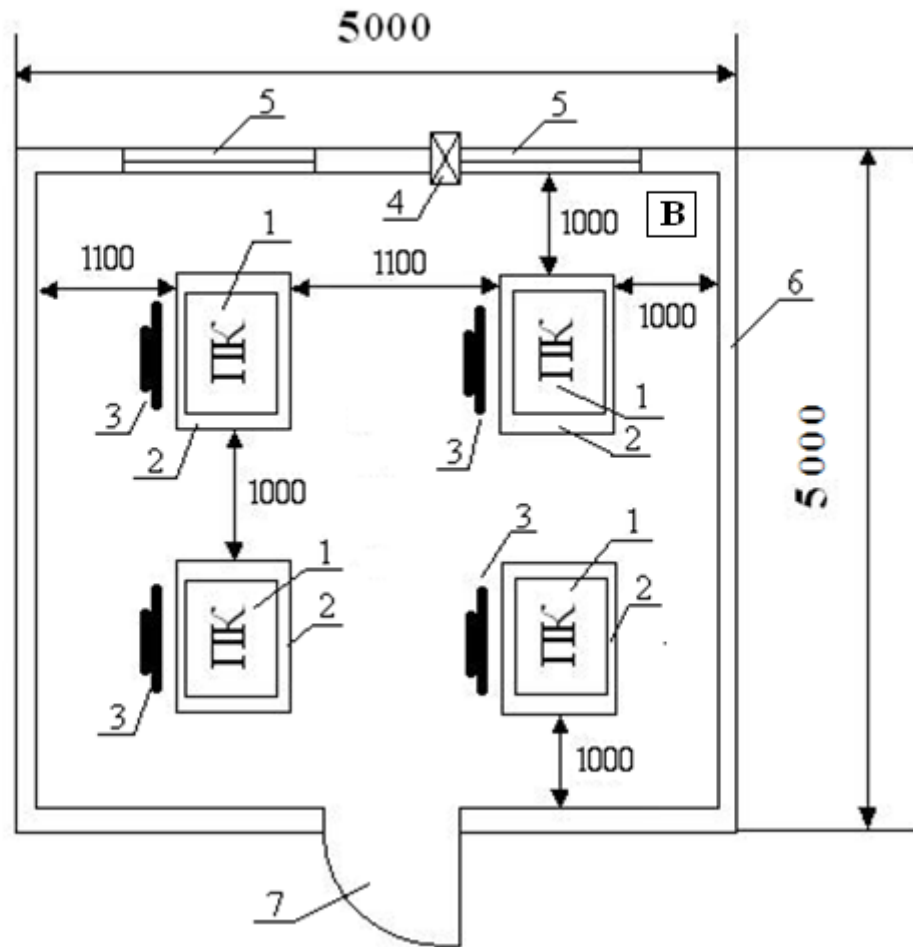


Рис.3.1. Схема організації робочих місць у приміщенні

Перелік предметів, що знаходяться у приміщенні, схема якого наведена на (рис.3.1.).

- 1 – Персональний комп'ютер;
- 2 – Стіл користувача;
- 3 – Крісло користувача;
- 4 – Кондиціонер;
- 5 – Вікно в приміщенні;
- 6 – Стіна;
- 7 – Двері.

3.3. Вимоги безпеки праці при роботі з комп'ютером

При виконанні робіт на комп'ютерах необхідно дотримуватись вимог загальної та даної інструкції з охорони праці.

До самостійної роботи з комп'ютерами допускаються особи, які пройшли навчання по професії, медичний огляд, вступний інструктаж з охорони праці, а також первинний інструктаж з охорони праці на робочому місці. В подальшому вони повинні проходити періодичні медичні огляди один раз на два роки, повторні інструктажі з охорони праці на робочому місці один раз на півріччя [12-13].

Під час роботи на комп'ютерах можуть діяти такі небезпечні та шкідливі фактори, як: фізичні, психофізіологічні. Основним обладнанням робочого місця користувача комп'ютера є монітор, системний блок та клавіатура.

Усі працівники, згідно статті 18 Закону України "Про охорону праці", зобов'язані [12]:

а) знати і виконувати вимоги нормативних актів про охорону праці, правила поведінки з устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту;

б) дотримуватись зобов'язань щодо охорони праці, передбачених колективним договором та правилами внутрішнього трудового розпорядку підприємства;

в) співробітничати з власником у справі організації безпечних і нешкідливих умов праці, особисто вживати посильних заходів щодо усунення будь-якої виробничої ситуації, яка створює загрозу його життю чи здоров'ю, або людей, які його оточують, повідомляти про небезпеку свого безпосереднього керівника або іншу посадову особу.

Основні правила користування комп'ютером [14]:

- монітор повинен розташовуватися так, щоб середина його була напроти очей людини, і розташовуватися від очей на відстані

витагнутої руки; на екрані не повинні відображатися ніякі предмети;

- вікно має бути розташоване перпендикулярно екрану монітора;
- освітленість в кімнаті повинна бути трохи менше освітленості екрану; для роботи з клавіатурою, літературою можливі додаткові освітлювачі, але вони не повинні давати відблиски на екран монітора;
- крісло має бути встановлено на оптимальній висоті, мати підлокітники, спинку за формою спини;
- клавіатура розташовується де завгодно, хоч на колінах, але так, щоб не треба було до неї тягнутися; те ж саме стосується і установки мишки:
- робочий стіл треба відрегулювати по висоті; він повинен бути досить вільним, щоб на ньому можна було розташувати всі комплектуючі, підставку під CD, підсвічування, місце для чашки чаю, літератури;
- не засиджуйтеся, необхідно весь час рухатися, частіше вставайте з місця, робіть розминки для рук, плечей, спини, очей тощо.

Правила техніки безпеки при роботі з комп'ютером [14]:

- перед початком роботи слід переконатися в справності електропроводки, вимикачів, розеток, за допомогою яких обладнання включається в мережу, працездатності комп'ютера;
- щоб уникнути пошкодження ізоляції електричних дротів і виникнення пошкоджень не дозволяється: вішати щось на дроти, зафарбовувати і білити шнури і дроти, класти проводи й дроти на газові чи водопровідні труби, за опалювальні системи, висмикувати вилку за шнур з розетки;
- забороняється під напругою очищати від пилу і забруднення електроприлади;

- забороняється перевіряти працездатність електроустаткування в непристосованих для експлуатації приміщеннях із струмопровідними підлогами;
- недозволено проводити ремонт засобів обчислювальної техніки і периферійного обладнання, ремонт електроапаратури, якщо вони знаходяться під напругою, а проводиться тільки відповідними фахівцями-техніками з дотриманням усіх відповідних вимог;
- щоб уникнути ураження електричним струмом, при користуванні електроприладами, не дозволяється торкатися будь-яких металевих конструкцій, з'єднаних з землею, а також трубопроводів, батарей опалення;
- при використанні електроприладів у сирих приміщеннях дотримуватися особливої обережності;
- після закінчення роботи необхідно знеструмити всі засоби обчислювальної техніки і периферійні обладнання, у разі безперервного виробничого процесу залишити тільки необхідне обладнання включеним.

Рекомендується: яскравість екрану монітора - не менше 100 кг/м²; контрастність зображення знаку - не менше 0,8; мінімальний розмір точки свічення для кольорового монітора не менше 0,6 мм; відношення яскравості монітора до яскравості оточуючих його поверхонь в робочій зоні - не більше 3:1 [14].

Якщо виявлено будь-яке пошкодження роботу не розпочинати, повідомити про це керівника.

3.4. Пожежна безпека приміщення

Особливо вірогідною і основною причиною заходження пожежі в приміщеннях експлуатації ПК є займання електропроводки через коротке замикання і незахищеність проводки.

Система попередження пожежі передбачає наступні заходи і засоби: надійне підключення електроустаткування до заземлюючих контуру без іскріння; дотримання режимів експлуатації на обладнанні згідно регламенту; наявність захисту від статичної і атмосферної електрики; навчання, атестація і переатестація персоналу з пожежної безпеки; періодично контроль цілісності і опору електроізоляції; своєчасне проведення профілактичних оглядів та ремонту.

ВИСНОВОК

В кваліфікаційній роботі досліджено роботу ФОП “Устимук В.С.”. В роботі була спроектована, розроблена та протестована система підтримки прийняття рішень логіста ФОП “Устимук В.С.”. Для створенної системи була створена інструкція користувача для даної інформаційної системи.

В результаті, створена система спрощує та прискорює прийняття рішення логістом, оскільки в ній зберігається необхідна для прийняття рішення інформація, а також, механізм розрахунку економічної вигідності та сигналізації про те чи вигідне замовлення, зменшує час, необхідний логістові.

Розроблена система дозволить скоротити час на обробку замовлень за допомогою автоматичного розрахунку економічної вигідності замовлень, а інформація про статус автомобіля та його технічні характеристики дозволить швидко обрати автомобіль для виконання серед автомобілів які зараз вільні, та підходять для виконання даного замовлення за вантажомісткістю.

Також система забезпечує низку додаткових функцій, таких як формування деяких звітів, які необхідні для звітності, та збереження інформації про замовників, та їх угоди.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ЛАРДИ-ТРАНС (LARDI-TRANS): ГРУЗЫ И ТРАНСПОРТ УКРАИНЫ [Електронний ресурс] – Режим доступу до ресурсу: <https://perevozky.com.ua/lardi-trans/>.
2. МЕТОДОЛОГІЯ IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: https://stud.com.ua/87184/ekonomika/metodologiya_idef0.
3. ПРОГРАМА 1С:ПІДПРИЄМСТВО 8 - ОСНОВНІ ВІДОМОСТІ І ХАРАКТЕРИСТИКИ [Електронний ресурс] – Режим доступу до ресурсу: <https://www.netsoft.com.ua/1C-Predpriyatiye-8-opisaniye-osnovnyye-kharakteristiki-ukr.html>.
4. Безкоштовні та "умовно безкоштовні" аналоги 1С:Підприємство в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://tqm.com.ua/ua/likbez/ua-articles/bezkoshtovni-analohy-1s>.
5. Microsoft Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio.
6. NuGet [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/NuGet>.
7. SQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/SQL>.
8. C Sharp [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/C_Sharp.
9. .NET Framework [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/.NET_Framework.
10. IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/IDEF0>.
11. Охорона праці [Електронний ресурс] – Режим доступу до ресурсу: <https://www.br.com.ua/diplom/technical/90033-23.html>.
12. Про охорону праці [Електронний ресурс] – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>.

13. Закон України про охорону праці [Електронний ресурс] – Режим доступу до ресурсу:
http://search.ligazakon.ua/l_doc2.nsf/link1/T269400.html.
14. Основные правила пользования компьютером. Правила работы за компьютером [Електронний ресурс] – Режим доступу до ресурсу:
<https://offlink.ru/bluetooth/basic-rules-for-using-a-computer-rules-of-work-at-the-computer/>.

ДОДАТКИ ДОДАТОК А

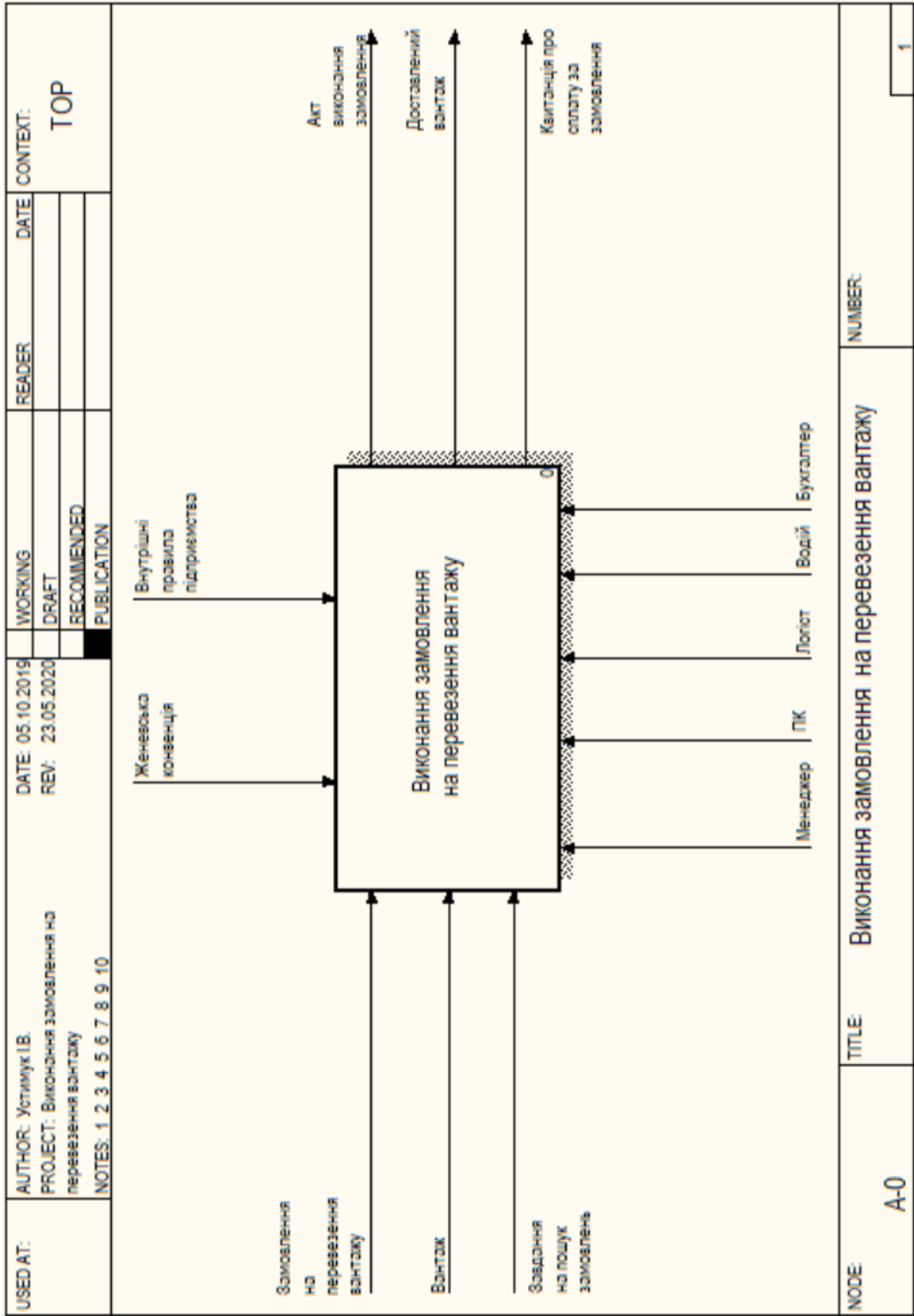


Рис А.1 Контекстна діаграма процесу виконання замовлення на перевезення підприємством

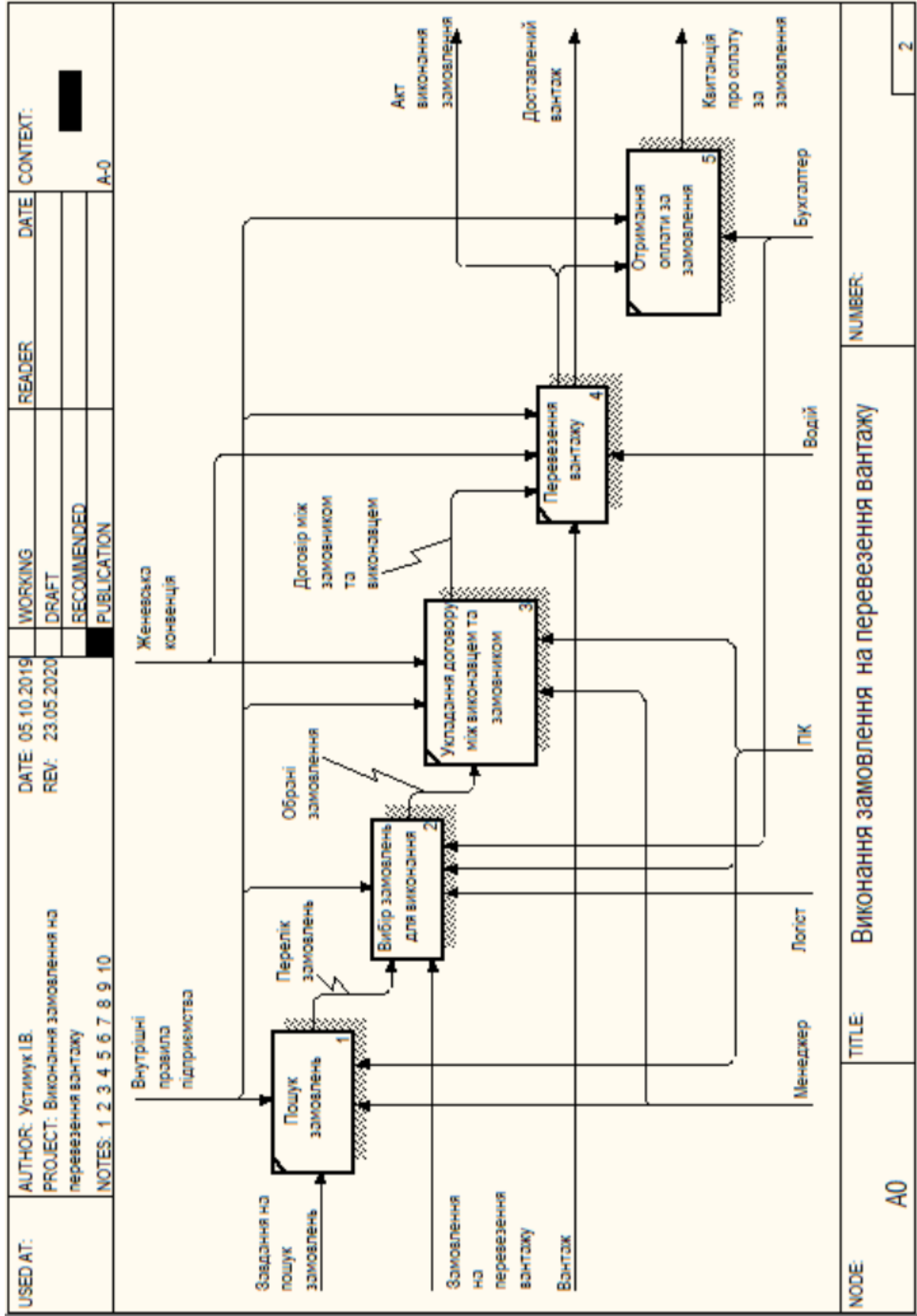


Рис А.2 Декомпозиція “Виконання замовлення на перевезення вантажу”

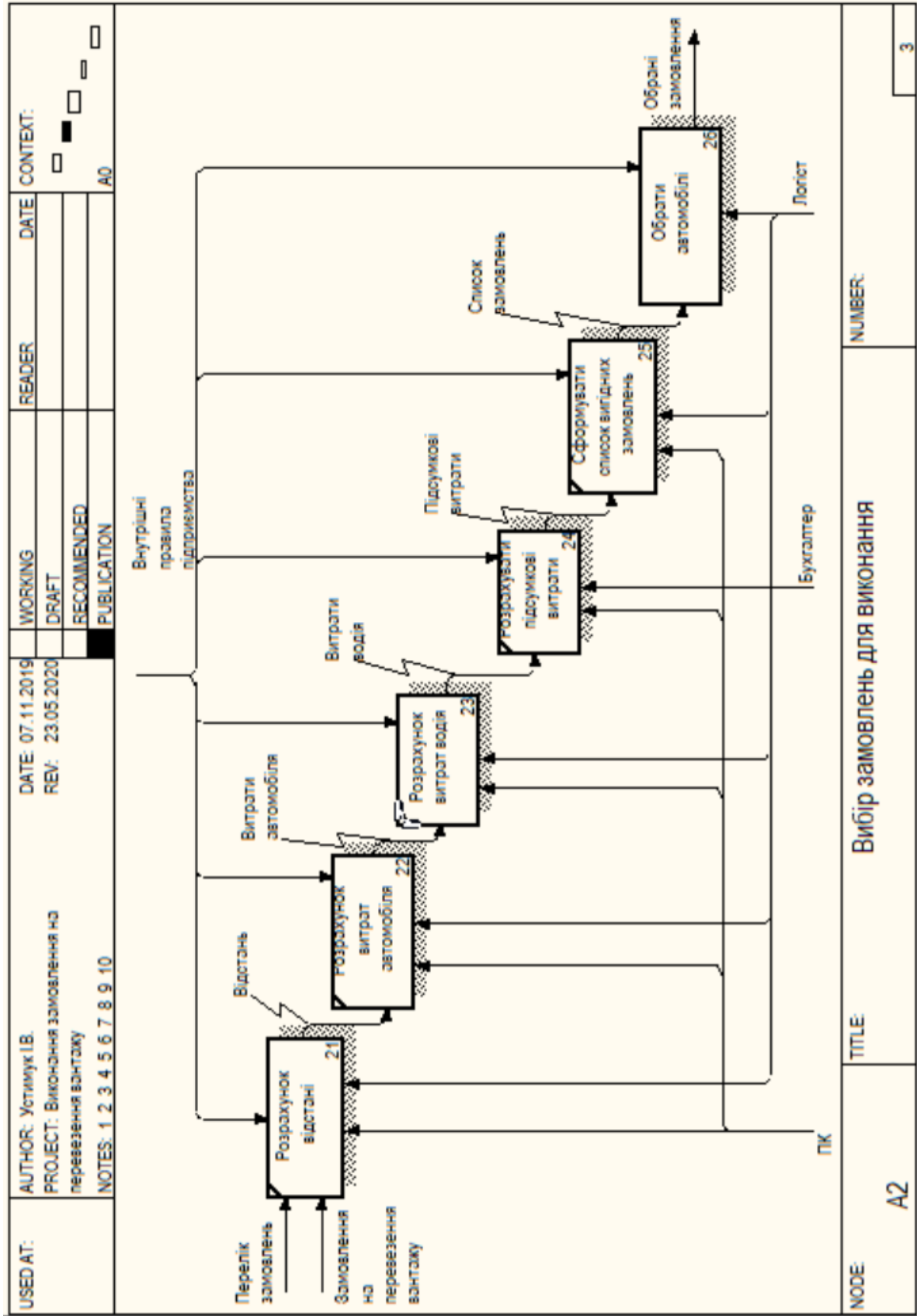


Рис А.3 Декомпозиція “Вибір замовлень для виконання”

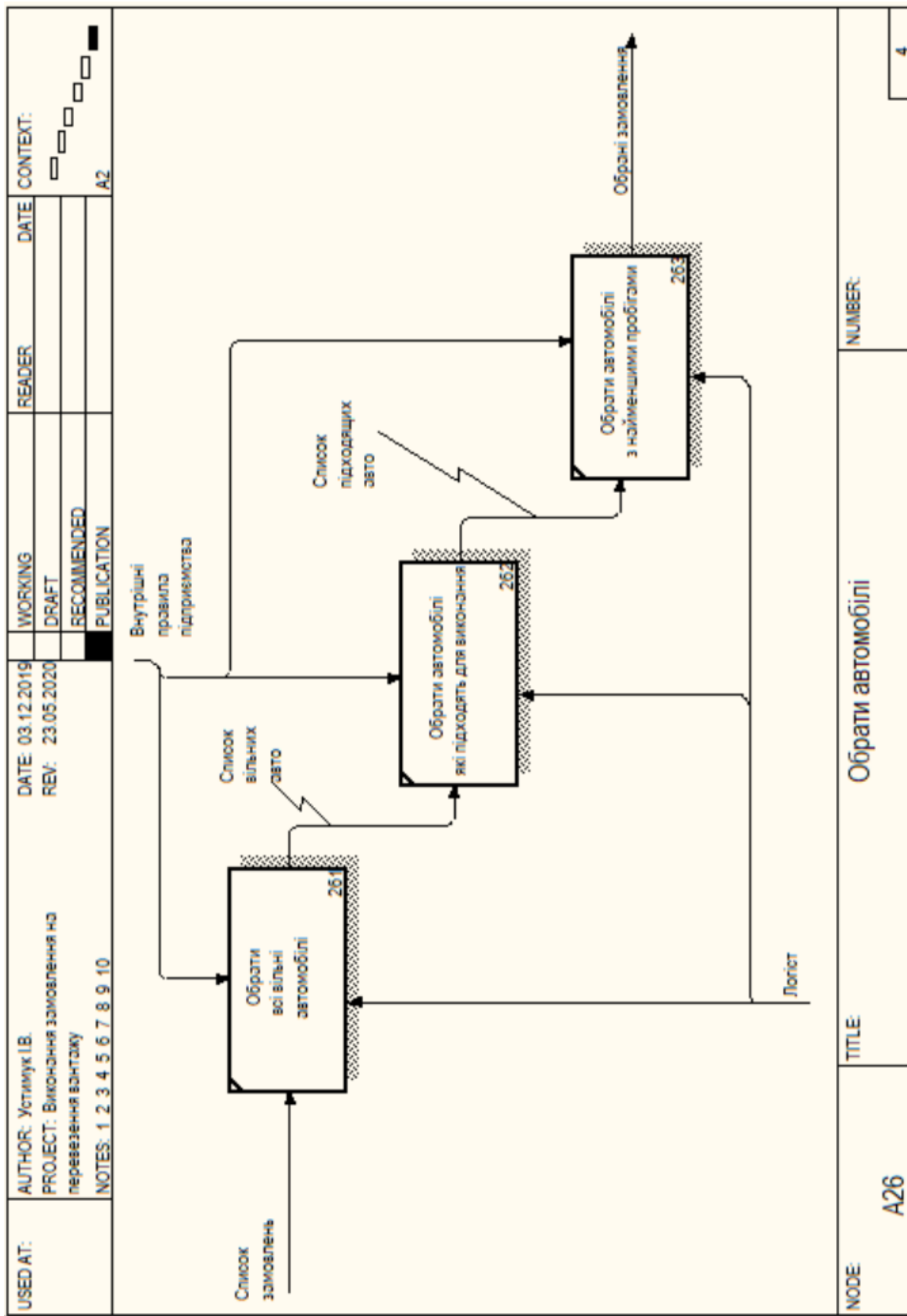


Рис А.4 Декомпозиція “Обрати автомобілі”

ДОДАТОК Б

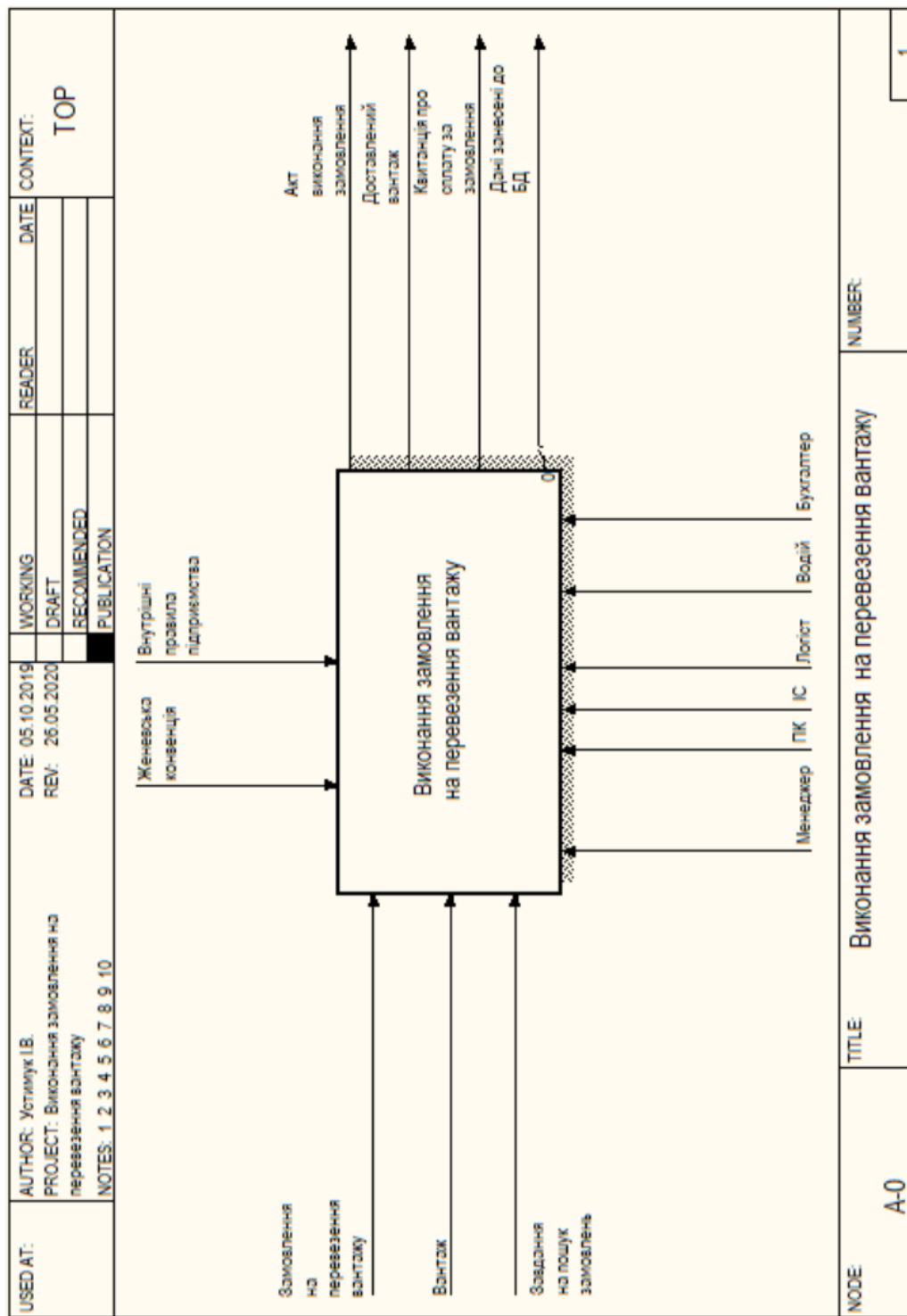


Рис Б.1 Концептуальна модель системи

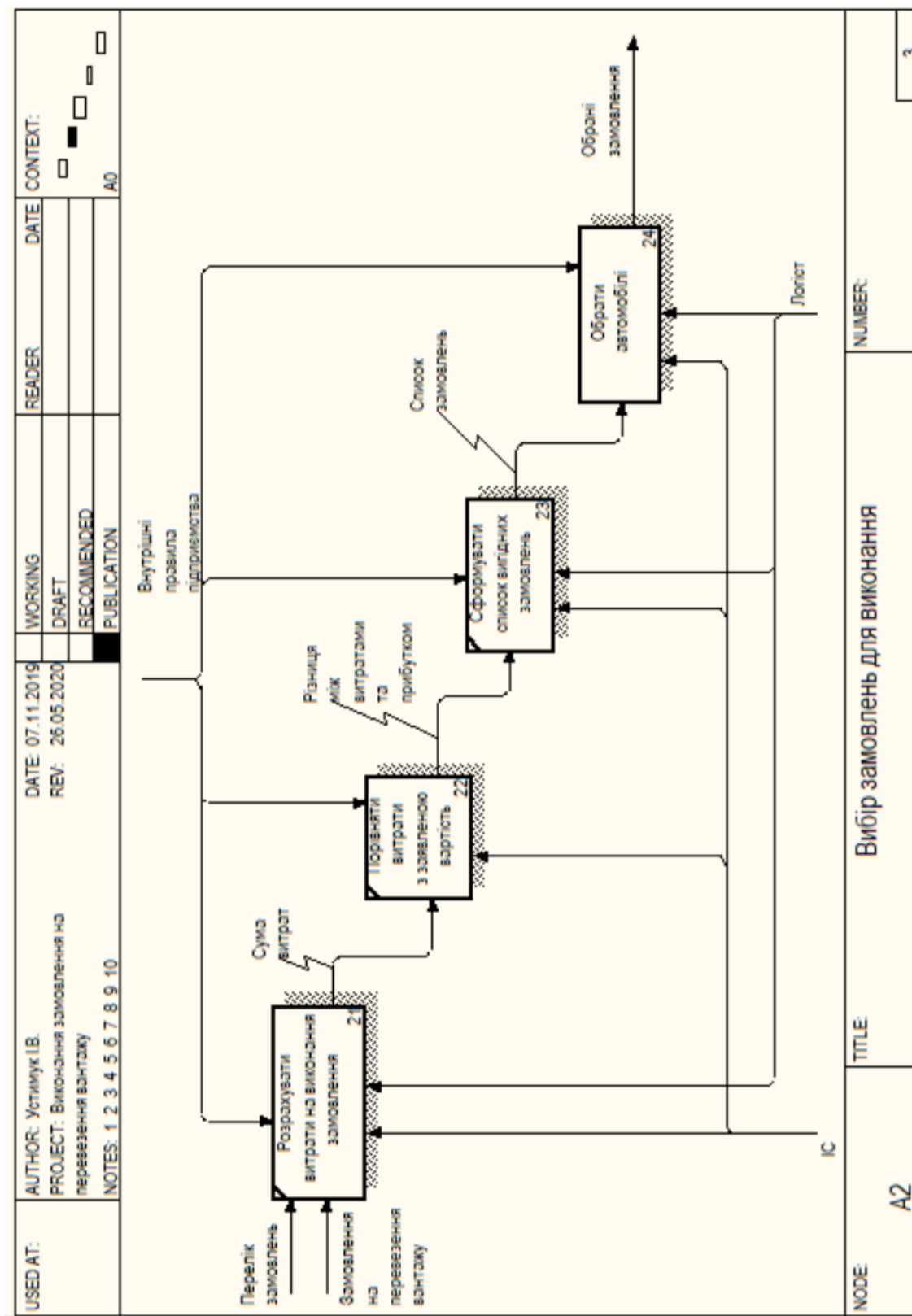


Рис Б.2 Декомпозиція “Вибір замовлень для виконання”(ТО-ВЕ)

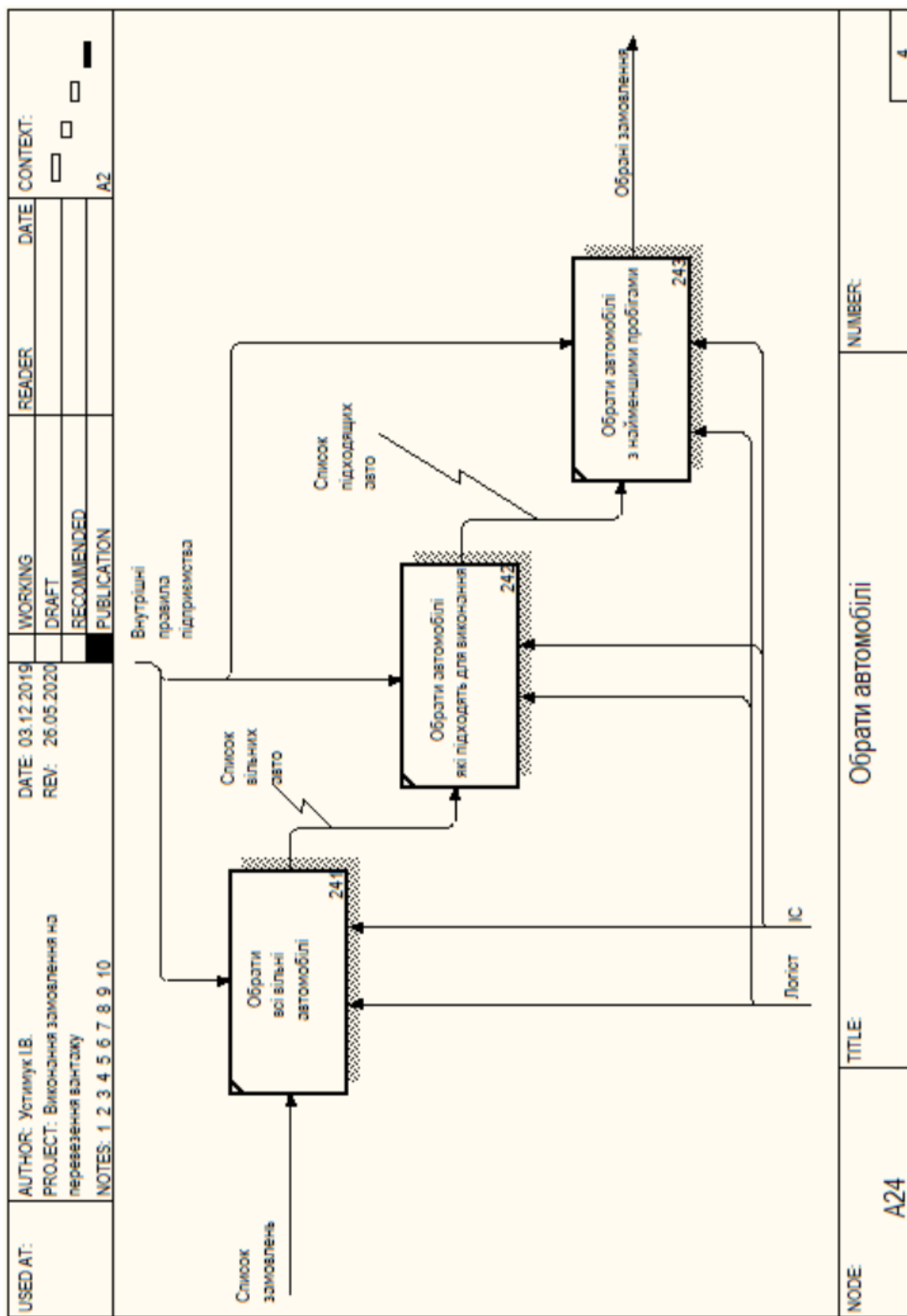


Рис Б.3 Декомпозиція “Обрати автомобілі” (TO-VE)

ДОДАТОК В

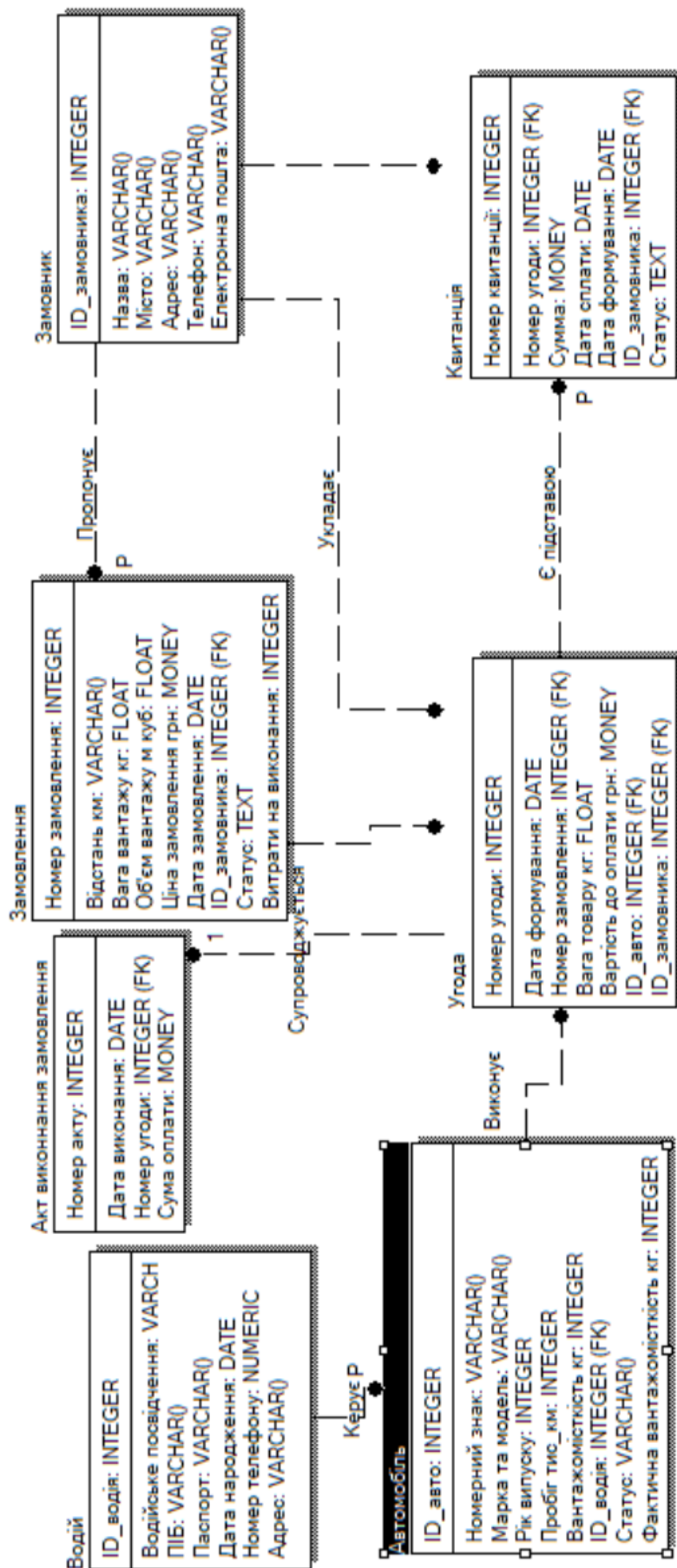


Рис В. Логічна схема БД

ДОДАТОК Г

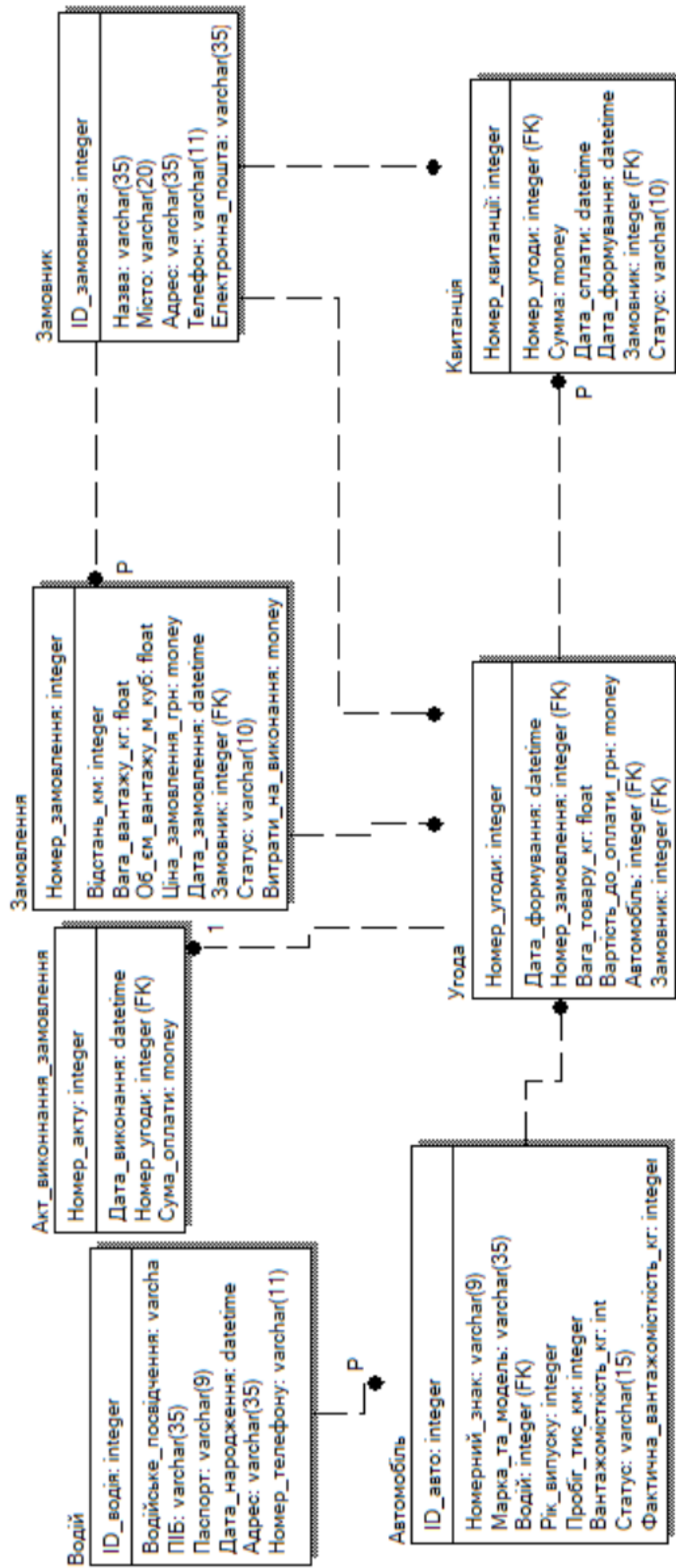


Рис Г. Фізична схема БД

ДОДАТОК Д. КОД ПРОГРАМИ

Примітка!!! Якщо в даному додатку відсутній код якогось вікна програми, це означає, що код цього вікна, абсолютно аналогічний до одного з вікон, код програми якого представлений в даному додатку.

А також, директиви, які підключаються, пропущені для всіх вікон окрім вікна довідника “Квитанція”, оскільки директиви скрізь аналогічні.

Довідник “Квитанція”

```
using MaterialSkin.Controls;
using System;
using System.Drawing;
using System.Linq;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
using System.ComponentModel;
using System.Data;
using System.Threading.Tasks;

namespace dip_2
{
    public partial class ticket : MaterialForm
    {
        public ticket()
        {
            InitializeComponent();
        }

        private void квитанціяBindingNavigatorSaveItem_Clickk(object sender, EventArgs e)
        {
            this.Validate();
        }
    }
}
```

```

        this.квитанціяBindin_Source.EndEdit();

        this.tableAdapterManager.UpdateAll(this.dIPDataSet);
    }

    private void ticket_Load(object sender, EventArgs e)
    {
        // TODO: "dIPDataSet.Замовник". .
        this.замовникTableAdapter.Fill(this.dIPDataSet.Замовник);

        // TODO: "dIPDataSet.Угода". .
        this.угодаTableAdapter.Fill(this.dIPDataSet.Угода);

        // TODO: "dIPDataSet.Квитанція". .
        this.квитанціяTableAdapter.Fill(this.dIPDataSet.Квитанція);
    }

    private void materialFlatButon1_Clickk(object sender, EventArgs e)
    {
        if (квитанціяDataGridView.Visible == false)
        {
            квитанціяDataGridView.Visible = true;
            materialFlatButon1.Text = "Приховати таблицю";
        }
        else
        {
            квитанціяDataGridView.Visible = false;
            materialFlatButon1.Text = "Показати таблицю";
        }
    }

    private void сумаTextBox_TextChanged(object sender, EventArgs e)
    {
        decimal value;
        decimal.TryParse(сумаTextBox.Text, out value);
        value = Math.Round(value, 2);
    }

```

```

        сумаTextBox.Text = value.ToString();
    }
}
}

```

Звіт “Замовлення клієнтів”

```

namespace dip_2
{
    public partial class report_order_client : MaterialForm
    {
        public report_order_client()
        {
            InitializeComponent();
        }

        private void report_order_client_Load(object sender, EventArgs e)
        {
            // TODO: "DIPDataSet.report_order_client". .
            this.report_order_clientTableAdapter.Fill(this.DIPDataSet.report_order_client);
            this.reportViewer1.RefreshReport();
        }
    }
}
}

```

Звіт “Угоди за обраний місяць”

```

namespace dip_2
{
    public partial class report_deals_by_month : MaterialForm
    {
        public report_deals_by_month()
        {
            InitializeComponent();
        }
    }
}

```



```

InitializeComponent();
}
private void materialRaisedButon1_Clickk(object sender, EventArgs e)
{
    try
    {
        this.заовлення_заовникаTableAdapter.Fill(this.dIPDataSet.заовлення_заовника,
        comboBox1.Text);
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}
private void order_by_client_Load(object sender, EventArgs e)
{
    // TODO: "dIPDataSet.Заовник". .
    this.заовникTableAdapter.Fill(this.dIPDataSet.Заовник);
}
private void materialFlatButon1_Clickk(object sender, EventArgs e)
{
    if (заовлення_заовникаDataGridView.Visible == false)
    {
        заовлення_заовникаDataGridView.Visible = true;
        materialFlatButon1.Text = "Приховати таблицю";
    }
    else
    {
        заовлення_заовникаDataGridView.Visible = false;
        materialFlatButon1.Text = "Показати таблицю";
    }
}

```

```

    }
    private void button2_Clickk(object sender, EventArgs e)
    {
        PrintDialog dlg = new PrintDialog();
        dlg.ShowDialog();
    }
    private void buton1_Clickk(object sender, EventArgs e)
    {
        if (замовлення_замовникаBindin_Source.Position + 1 <
замовлення_замовникаBindin_Source.Count)
            замовлення_замовникаBindin_Source.MoveNext();
    }
    private void button5_Clickk(object sender, EventArgs e)
    {
        if (замовлення_замовникаBindin_Source.Position - 1 <= 1)
            замовлення_замовникаBindin_Source.MovePrevious();
    }
}
}
}

```

Довідник “Замовлення”

```

namespace dip_2
{
    public partial class order : MaterialForm
    {
        public order()
        {
            InitializeComponent();
        }
        private void замовленняBindingNavigatorSaveItem_Clickk(object sender, EventArgs e)
        {
            this.Validate();
        }
    }
}

```

```

this.заовленняBindin_Source.EndEdit();

try
{
    this.tableAdapterManager.UpdateAll(this.dIPDataSet);
}
catch (SqlException ex)
{
    MessageBox.Show(" ____ Помилка __: " + ex.Message, " ____ Помилка __");
}
}

private void order_Load(object sender, EventArgs e)
{
    // TODO: "dIPDataSet.Заовлення". .
    this.заовленняTableAdapter.Fill(this.dIPDataSet.Заовлення);
    // TODO: "dIPDataSet.Заовник". .
    this.заовникTableAdapter.Fill(this.dIPDataSet.Заовник);
    // TODO: "dIPDataSet.Заовлення". .
    this.заовленняTableAdapter.Fill(this.dIPDataSet.Заовлення);

}

private void materialFlatButon1_Clickk(object sender, EventArgs e)
{
    if (заовленняDataGridView.Visible == false)
    {
        заовленняDataGridView.Visible = true;
        materialFlatButon1.Text = "Приховати таблицю";
    }
    else
    {
        заовленняDataGridView.Visible = false;
    }
}

```

```

        materialFlatButton1.Text = "Показати таблицю";
    }
}
private void ціна_замовлення_грнTextBox_TextChanged(object sender, EventArgs e)
{
    decimal value;
    decimal.TryParse(ціна_замовлення_грнTextBox.Text, out value);
    value = Math.Round(value, 2);
    ціна_замовлення_грнTextBox.Text = value.ToString();
}
private void витрати_на_виконанняTextBox_TextChanged(object sender, EventArgs e)
{
    витрати_на_виконанняTextBox.BackColor = Color.White;
    decimal value=0,value1;
    decimal.TryParse(витрати_на_виконанняTextBox.Text, out value);
    value = Math.Round(value, 2);
    витрати_на_виконанняTextBox.Text = value.ToString();
    if (value>0)
    {
        decimal.TryParse(ціна_замовлення_грнTextBox.Text, out value1);
        if (value >= value1 - 5000)
            витрати_на_виконанняTextBox.BackColor = Color.Red;
        else
            витрати_на_виконанняTextBox.BackColor = Color.Green;
    }
}
private void materialCheckBox1_CheckedChanged(object sender, EventArgs e)
{
    if (materialCheckBox1.Checked == true)
    {

```

```

        this.замовленняBindin_Source.Filter = "Статус IS NULL";
    }
    else
    {
        this.замовленняBindin_Source.RemoveFilter();
    }
}
private void замовленняBindingNavigatorSaveItem_Clickk_1(object sender, EventArgs e)
{
    this.Validate();
    this.замовленняBindin_Source.EndEdit();
    this.tableAdapterManager.UpdateAll(this.dIPDataSet);
}
private void materialRaisedButon1_Clickk(object sender, EventArgs e)
{
    deal N = new dip_2.deal();
    N.ShowDialog();
}
}
}
}

```

Головне вікно програми

```

namespace dip_2
{
    public partial class Form1 : MaterialForm
    {
        public Form1()
        {
            InitializeComponent();
            timer1.Start();
            acssess_form N = new acssess_form();

```

```

        N.ShowDialog();

    }

    private void довідникВодійToolStripMenuItem_Clickk(object sender, EventArgs e)
    {
        driver N = new driver();
        N.ShowDialog();
    }

    private void довідникАвтомобільToolStripMenuItem_Clickk(object sender, EventArgs
e)
    {
        car N = new dip_2.car();
        N.ShowDialog();
    }

    private void довідникЗамовникToolStripMenuItem_Clickk(object sender, EventArgs e)
    {
        client N = new client();
        N.ShowDialog();
    }

    private void довідникЗамовленняToolStripMenuItem_Clickk(object sender, EventArgs
e)
    {
        order N = new order();
        N.ShowDialog();
    }

    private void довідникУгодаToolStripMenuItem_Clickk(object sender, EventArgs e)
    {
        deal N = new deal();
        N.ShowDialog();
    }

    private void довідникАктВиконанихРобітToolStripMenuItem_Clickk(object sender,
EventArgs e)

```

```

    {
        act_of_done_work N = new act_of_done_work();
        N.ShowDialog();
    }

private void довідникКвитанціяToolStripMenuItem_Clickk(object sender, EventArgs
e)
    {
        ticket N = new ticket();
        N.ShowDialog();
    }

private void пошукЗамовленняЗамовникаToolStripMenuItem_Clickk(object sender,
EventArgs e)
    {
        order_by_client N = new order_by_client();
        N.ShowDialog();
    }

private void пошукУгодиЗамовникаToolStripMenuItem_Clickk(object sender,
EventArgs e)
    {
        deal_by_client N = new deal_by_client();
        N.ShowDialog();
    }

private void пошукАвтомобіліВантажомісткостіToolStripMenuItem_Clickk(object
sender, EventArgs e)
    {
        car_max_mass N = new car_max_mass();
        N.ShowDialog();
    }

private void toolStripMenuItem1_Clickk(object sender, EventArgs e)
    {
        car_driver_numb N = new car_driver_numb();
        N.ShowDialog();
    }

```

```

    }

    private void
пошукАктВиконанихРобітТаКвитанціяУгодиToolStripMenuItem_Clickk(object sender,
EventArgs e)
    {
        act_ticket_of_deal N = new act_ticket_of_deal();
        N.ShowDialog();
    }

    private void звітУгодиЗамовникаToolStripMenuItem_Clickk(object sender, EventArgs
e)
    {
        report_deals_client N = new report_deals_client();
        N.ShowDialog();
    }

    private void toolStripMenuItem2_Clickk(object sender, EventArgs e)
    {
        report_deals_by_month N = new report_deals_by_month();
        N.ShowDialog();
    }

    private void звітУгодиАвтомобілівToolStripMenuItem_Clickk(object sender,
EventArgs e)
    {
        report_cars_deals N = new report_cars_deals();
        N.ShowDialog();
    }

    private void звітЗамовленняЗамовниківToolStripMenuItem_Clickk(object sender,
EventArgs e)
    {
        report_order_client N = new report_order_client();
        N.ShowDialog();
    }

    private void закритиПрограмуToolStripMenuItem_Clickk(object sender, EventArgs e)

```

```

    {
        Application.Exit();
    }

    private void вийтиЗОбліковогоЗаписуToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        access_form N = new access_form();
        N.ShowDialog();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

        //start timer when form loads
        timer1.Start(); //this will use t_Tick() method
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        label1.Text = DateTime.Now.ToString("HH:mm");
        label2.Text = DateTime.Now.ToString("ss");
        label3.Text = DateTime.Now.ToString("MMM dd yyyy");
        label4.Text = DateTime.Now.ToString("dddd");
    }
}
}
}

```

Довідник “Водій”

```

namespace dip_2
{
    public partial class driver : MaterialForm
    {
        public driver()
        {

```

```

InitializeComponent();
}
private void водійBindingNavigatorSaveItem_Clickk(object sender, EventArgs e)
{
    this.Validate();
    this.водійBindin_Source.EndEdit();
    this.tableAdapterManager.UpdateAll(this.dIPDataSet);

}
private void driver_Load(object sender, EventArgs e)
{
    // TODO: "dIPDataSet.Водій". .
    this.водійTableAdapter.Fill(this.dIPDataSet.Водій);

}
private void materialFlatButon1_Clickk(object sender, EventArgs e)
{
    if(водійDataGridView.Visible==false)
    {
        водійDataGridView.Visible = true;
        materialFlatButon1.Text = "Приховати таблицю";
    }
    else
    {
        водійDataGridView.Visible = false;
        materialFlatButon1.Text = "Показати таблицю";
    }
}
private void button2_Clickk(object sender, EventArgs e)
{

```

```

        this.Validate();
        this.водійBindin_Source.EndEdit();
        this.tableAdapterManager.UpdateAll(this.dIPDataSet);
    }
    private void button1_Clickk(object sender, EventArgs e)
    {
        if (водійBindin_Source.Position + 1 < водійBindin_Source.Count)
            водійBindin_Source.MoveNext();
    }
    private void button5_Clickk(object sender, EventArgs e)
    {
        if (водійBindin_Source.Position - 1 <= 1)
            водійBindin_Source.MovePrevious();
    }
    private void button4_Clickk(object sender, EventArgs e)
    {
        водійBindin_Source.RemoveCurrent();
    }
    private void button3_Clickk(object sender, EventArgs e)
    {
        водійBindin_Source.AddNew();
    }
    private void materialFlatButton2_Clickk(object sender, EventArgs e)
    {
        if (materialSingleLineTextField1.Text.Length == 0)
            MessageBox.Show("Некоректно введені дані для пошуку", " ____ Помилка ____");
        else
        {
            int itemFound = this.водійBindin_Source.Find("Паспорт",
materialSingleLineTextField1.Text);
            this.водійBindin_Source.Position = itemFound;
        }
    }

```

```

    }
}
private void materialFlatButton3_Clickk(object sender, EventArgs e)
{
    if (materialLabel3.Visible == false)
    {
        this.водійBindin_Source.Filter = "Паспорт LIKE%" +
materialSingleLineTextField1.Text + "%";
        materialLabel3.BackColor = Color.Red;
        materialLabel3.Visible = true;
    }
    else
    {
        this.водійBindin_Source.RemoveFilter();
        materialLabel3.Visible = false;
    }
}
private void materialFlatButton4_Clickk(object sender, EventArgs e)
{
    car N = new car();
    N.ShowDialog();
}
}
}

```

Довідник “Угода”

```

namespace dip_2
{
    public partial class deal : MaterialForm
    {
        public deal()
        {

```

```

InitializeComponent();
}
private void угодаBindingNavigatorSaveItem_Clickk(object sender, EventArgs e)
{
    this.Validate();
    this.угодаBindin_Source.EndEdit();
    this.tableAdapterManager.UpdateAll(this.dIPDataSet);
}
private void deal_Load(object sender, EventArgs e)
{
    // TODO: "dIPDataSet.Автомобіль1". .
    this.автомобіль1TableAdapter.Fill(this.dIPDataSet.Автомобіль1);
    // TODO: "dIPDataSet.report_cars_deals". .
    // TODO: "dIPDataSet.Замовник". .
    this.замовникTableAdapter.Fill(this.dIPDataSet.Замовник);
    // TODO: "dIPDataSet.Замовлення". .
    this.замовленняTableAdapter.Fill(this.dIPDataSet.Замовлення);
    // TODO: "dIPDataSet.Угода". .
    this.угодаTableAdapter.Fill(this.dIPDataSet.Угода);
}
private void materialFlatButon1_Clickk(object sender, EventArgs e)
{
    if (угодаDataGridView.Visible == false)
    {
        угодаDataGridView.Visible = true;
        materialFlatButon1.Text = "Приховати таблицю";
    }
    else
    {

```

```

        угодаDataGridView.Visible = false;
        materialFlatButon1.Text = "Показати таблицю";
    }
}
private void вартість_до_оплати_грнTextBox_TextChanged(object sender, EventArgs e)
{
    decimal value;
    decimal.TryParse(вартість_до_оплати_грнTextBox.Text, out value);
    value = Math.Round(value, 2);
    вартість_до_оплати_грнTextBox.Text = value.ToString();
}
private void materialRaisedButon1_Clickk(object sender, EventArgs e)
{
    ticket N = new ticket();
    N.ShowDialog();
}
private void materialRaisedButton2_Clickk(object sender, EventArgs e)
{
    act_of_done_work N = new act_of_done_work();
    N.ShowDialog();
}
}
}
}

```

Довідник “Автомобіль”

```

namespace dip_2
{
    public partial class car : MaterialForm
    {
        public car()
        {

```

```

        InitializeComponent();
    }
    private void автомобільBindingNavigatorSaveItem_Clickk(object sender, EventArgs e)
    {
        this.Validate();
        this.автомобільBindin_Source.EndEdit();
        this.tableAdapterManager.UpdateAll(this.dIPDataSet);
    }
    private void car_Load(object sender, EventArgs e)
    {
        // TODO: "dIPDataSet.Водій". .
        this.водійTableAdapter.Fill(this.dIPDataSet.Водій);
        // TODO: "dIPDataSet.Автомобіль". .
        this.автомобільTableAdapter.Fill(this.dIPDataSet.Автомобіль);
        // TODO: "dIPDataSet.Водій". .
        this.водійTableAdapter.Fill(this.dIPDataSet.Водій);
        // TODO: "dIPDataSet.Автомобіль". .
        this.автомобільTableAdapter.Fill(this.dIPDataSet.Автомобіль);
    }
    private void materialFlatButon1_Clickk(object sender, EventArgs e)
    {
        if (автомобільDataGridView.Visible == false)
        {
            автомобільDataGridView.Visible = true;
            materialFlatButon1.Text = "Приховати таблицю";
        }
        else
        {
            автомобільDataGridView.Visible = false;
        }
    }

```



```

else
    if(materialSingleLineTextField1.Text.Length==0)
        MessageBox.Show("Некоректно введені дані для пошуку", " ____ Помилка
__");
    else
        this.автомобільBindin_Source.Position = itemFound;
}
}
private void button3_Clickk_1(object sender, EventArgs e)
{
    автомобільBindin_Source.AddNew();
}
private void автомобільBindingNavigatorSaveItem_Clickk_1(object sender, EventArgs
e)
{
    this.Validate();
    this.автомобільBindin_Source.EndEdit();
    this.tableAdapterManager.UpdateAll(this.dIPDataSet);
}
private void button2_Clickk_1(object sender, EventArgs e)
{
    this.Validate();
    this.автомобільBindin_Source.EndEdit();
    this.tableAdapterManager.UpdateAll(this.dIPDataSet);
}
private void button4_Clickk_1(object sender, EventArgs e)
{
    автомобільBindin_Source.RemoveCurrent();
}
private void button5_Clickk_1(object sender, EventArgs e)

```

```

{
    if (автомобільBindin_Source.Position - 1 <= 1)
        автомобільBindin_Source.MovePrevious();
}

privte void materialFlatButton3_Clickk(object sender, EventArgs e)
{
    if (materialLabel3.Visible == false)
    {
        int value=0;
        string param = "";
        if (materialRadioButon1.Checked == true)
        {
            param = "Номерний_знак";
            this.автомобільBindin_Source.Filter = param + " LIKE%" +
materialSingleLineTextField1.Text + "%";
        }
        if (materialRadioButton2.Checked == true)
        {
            param = "Рік_випуску";
            int.TryParse(materialSingleLineTextField1.Text, out value);
            this.автомобільBindin_Source.Filter = param + " =" + value;
        }
        if (materialRadioButton3.Checked == true)
        {
            param = "Пробіг_тис_км";
            int.TryParse(materialSingleLineTextField1.Text, out value);
            this.автомобільBindin_Source.Filter = param + " =" + value;
        }
        if (materialRadioButton4.Checked == true)
        {
            param = "Вантажомісткість_кг";

```

```

        int.TryParse(materialSingleLineTextField1.Text, out value);
        this.автомобільBindin_Source.Filter = param + " = " + value;
    }
    if (param == "")
        MessageBox.Show("Оберіть критерій для фільтрації", " ____ Помилка ____");
    else
    {
        materialLabel3.BackColor = Color.Red;
        materialLabel3.Visible = true;
    }
}
else
{
    this.автомобільBindin_Source.RemoveFilter();
    materialLabel3.Visible = false;
}
}
private void materialCheckBox1_CheckedChanged(object sender, EventArgs e)
{
    if (materialCheckBox1.Checked == true)
    {
        this.автомобільBindin_Source.Filter = "Статус LIKE '%Вільн%'";
    }
    else
    {
        this.автомобільBindin_Source.RemoveFilter();
    }
}
}
}
}

```

Вікно входу до системи

```
namespace dip_2
{
    public partial class access_form : MaterialForm
    {
        public access_form()
        {
            InitializeComponent();
        }

        private void materialRaisedButon1_Clickk(object sender, EventArgs e)
        {
            if(Control.IsKeyLocked(Keys.CapsLock)==true)
            {
                pictureBox1.Visible = true;
                materialLabel3.Visible = true;
            }
            else
            {
                if(materialLabel3.Visible = true)
                {
                    pictureBox1.Visible = false;
                    materialLabel3.Visible = false;
                }

                if (materialSingleLineTextField1.Text == "admin" &&
                    materialSingleLineTextField2.Text == "admin")
                    Close();
            }
            else
            {
                MessageBox.Show("Введений логін або пароль невірний, будь ласка спробуйте ще раз", " __ Помилка __");
                materialSingleLineTextField1.Text = "";
            }
        }
    }
}
```

```
        materialSingleLineTextField2.Text = "";  
    }  
}  
}  
}
```