

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеня  
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Часника Владислава Сергійовича*  
(ПІБ)

академічної групи *122-19ск-2*  
(шифр)

спеціальності *122 Комп'ютерні науки*  
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*  
(назва освітньої програми)

на тему: *Розробка інформаційної автоматизованої системи  
для роботи диспетчера сервісу оренди автотранспорту  
для вантажоперевезень*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Кабак Л.В.</i>			
<b>розділів:</b>				
спеціальний	<i>доц. Кабак Л.В.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
<b>Рецензент</b>	<i>доц. Шедловський І.А.</i>			
<b>Нормоконтролер</b>	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2022

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

«    »                      2022 року

**ЗАВДАННЯ**

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-19ск-2 Часника Владислава Сергійовича  
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка інформаційної автоматизованої системи для роботи диспетчера сервісу оренди автотранспорту для вантажоперевезень

затверджена наказом ректора НТУ «ДП» від «18» травня 2022 р. № 268-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів практик та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	13.05.2022 р.
Економічний	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	27.05.2022 р.

Завдання видав

(підпис)

доц. Кабак Л.В.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Часник В.С.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

## РЕФЕРАТ

Пояснювальна записка: 77 с., 22 рис., 5 табл., 3 дод., 20 джерел.

Об'єкт розробки: інформаційна автоматизована система для роботи диспетчера сервісу оренди автотранспорту для вантажоперевезень.

Мета кваліфікаційної роботи: розробка та оптимізація роботи автоматизованого робочого місця диспетчера сервісу оренди автотранспорту. Полегшення процесу роботи з клієнтами, співробітниками, автомобілями та формування контрактів за рахунок розробки та використанню відповідного додатку.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної області, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування підсистеми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження застосунку, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню застосунку та розраховано час на його створення.

Практичне значення полягає в розробці та оптимізації роботи автоматизованого робочого місця диспетчера сервісу оренди автотранспорту..

Актуальність даного типу додатків визначається великим попитом на подібні розробки.

Список ключових слів: ОRENDA, ДОДАТОК, КЛІЄНТ, ВОДІЙ, ДОГОВІР, АВТОТРАНСПОРТ.

## **ABSTRACT**

Explanatory note: 77 pp., 22 fig., 5 tabs., 3 apps, 20sources.

Object of development: information automated system for the work of the dispatcher of the service of rent of motor transport for cargo transportation.

The purpose of the qualification work: development and optimization of the automated workplace of the dispatcher of the car rental service. Facilitate the process of working with customers, employees, cars and the formation of contracts through the development and use of appropriate applications.

The introduction considers the analysis and current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development are defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes the existing solutions, selects the platform for development, performs design and development of the program, describes the algorithm and structure of the subsystem, determines the input and output data, provides characteristics of the parameters of technical means, describes the call and application load, describes the program.

The economic section determines the complexity of the developed information system, calculates the cost of work to create an application and calculates the time to create it.

The practical significance lies in the development and optimization of the automated workplace of the dispatcher of the car rental service.

The relevance of this type of application is determined by the high demand for such developments.

Keywords: RENT, APPLICATION, CUSTOMER, DRIVER, CONTRACT, VEHICLES.

## ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Загальні відомості з предметної галузі.....	9
1.2 Призначення розробки та область застосування.....	11
1.3 Підстава для розробки.....	11
1.4 Постановка завдання.....	12
1.5 Вимоги до програми або програмного виробу.....	13
1.5.1 Вимоги до функціональних характеристик .....	13
1.5.2 Вимоги до інформаційної безпеки.....	14
1.5.3 Вимоги до складу та параметрів технічних засобів.....	15
1.5.4 Вимоги до інформаційної та програмної сумісності.....	16
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	17
2.1 Функціональне призначення системи.....	17
2.2 Опис застосованих математичних методів.....	18
2.3 Опис використаних технологій та мов програмування.....	18
2.4 Опис структури системи та алгоритмів її функціонування.....	21
2.5 Обґрунтування та організація вхідних та вихідних даних програми.....	31
2.6 Опис роботи розробленої системи.....	31
2.6.1 Використані технічні засоби.....	31
2.6.2 Використані програмні засоби.....	32
2.6.3 Виклик та завантаження програми.....	32

2.6.4	Опис інтерфейсу користувача.....	32
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....		42
3.1	Розрахунок трудомісткості та вартості розробки програмного продукту .....	42
3.2	Розрахунок витрат на створення програми.....	46
ВИСНОВКИ.....		48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		49
Додаток А. Код програми.....		50
Додаток Б. Відгук керівника економічного розділу.....		76
Додаток В. Перелік файлів на диску.....		77

## СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

- БД - база даних;
- ЕОМ - електронно-обчислювальна машина;
- ІС - інформаційна система;
- ПЗ - програмне забезпечення;
- СУБД - система управління базами даних;
- ІТ - інформаційні технології.

## ВСТУП

Зростаючі темпи інформатизації суспільства сприяють підвищенню ролі обчислювальної техніки в управлінських процесах. Можливості сучасної обчислювальної техніки для автоматизації процесу обробки інформації дозволяють збільшити продуктивність праці, підвищити ефективність роботи з документами і прискорити обмін управлінською інформацією.

В даний час велике поширення набула концепція розподілених автоматизованих систем управління, спрямованих на локальну обробку інформації. Це дозволяє організувати поділ праці управлінського персоналу та автоматизувати виконання ним своїх функцій. Для реалізації даної ідеї необхідно створити для кожного рівня управління і кожної предметної області автоматизовані робочі місця на базі персональних комп'ютерів.

Автоматизоване робоче місце (АРМ) — це комплекс засобів обчислювальної техніки і програмного забезпечення, що розташовується безпосередньо на робочому місці співробітника і призначений для автоматизації його роботи в рамках спеціальності.

Оренда автотранспорту для вантажоперевезень у наш час набуває все більшої популярності. Малі та середні компанії не можуть собі дозволити утримувати вантажну техніку, тому звертаються до сервісів оренди. Окремим людям також можуть знадобитися послуги вантажоперевезення для власних потреб, тому розробка у найближчий час не втратить своєї актуальності.

Метою кваліфікаційної роботи є розробка та оптимізація роботи автоматизованого робочого місця диспетчера сервісу оренди автотранспорту. Полегшення процесу роботи з клієнтами, співробітниками, автомобілями та формування контрактів за рахунок розробки та використанню відповідного додатку.



# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1. Загальні відомості з предметної галузі

Найближчим по функціоналу програмним продуктом є програма «AutoSoft». Програма для обліку автотранспортної техніки власного парку, документів, пов'язаних з обліком, формування та друку подорожніх листів, обліку виконаних ремонтних робіт і ТО, обліку ПММ, роботи водіїв, обліку контрагентів і роботи з ними, складського обліку. Інтерфейс програми додавання нового автотранспорту представлено на рис. 1.1.

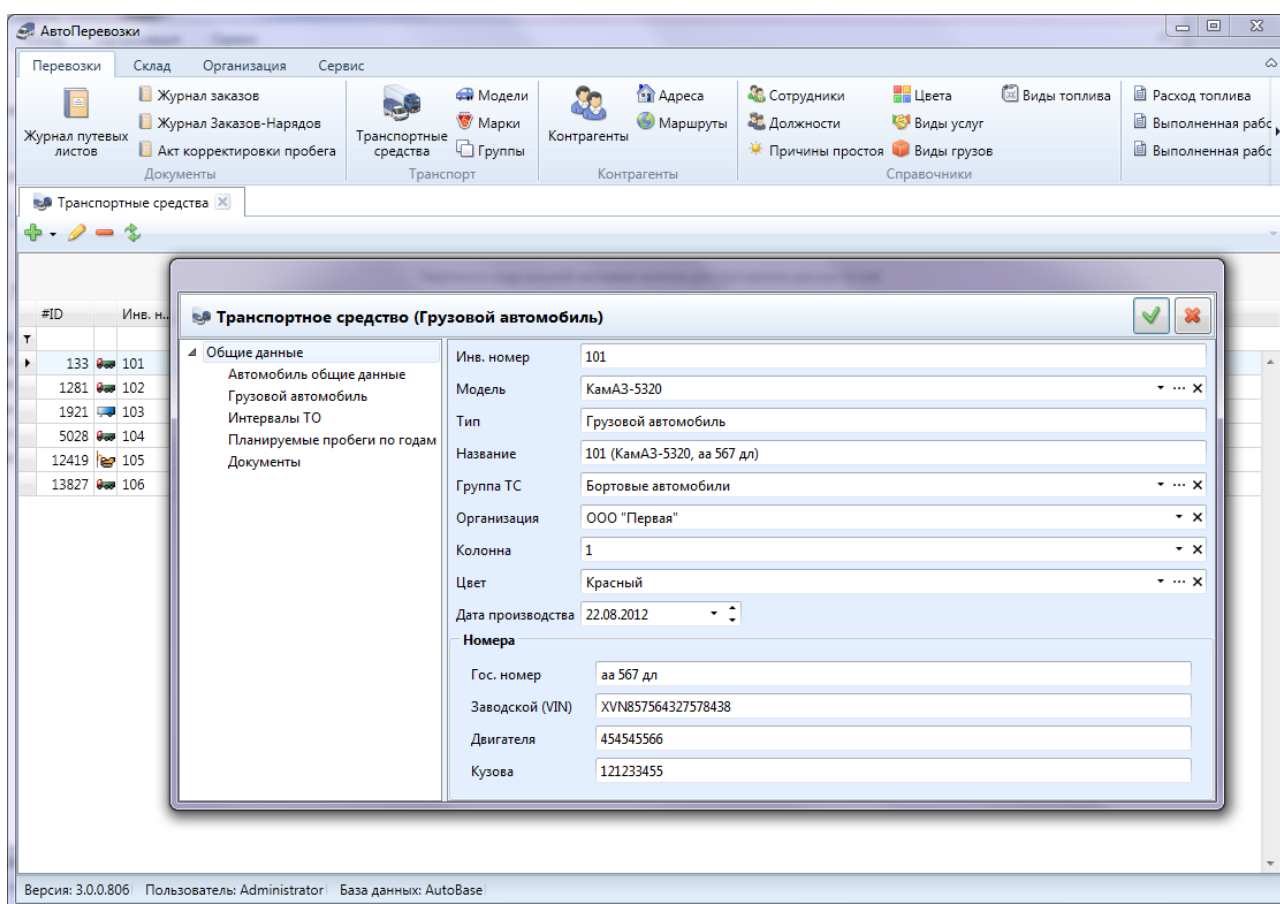


Рис. 1.1. Интерфейс программы «AutoSoft»

Програма «AutoPlan». Роботизований програмний комплекс для автоматизації обліку і управління на автотранспортному підприємстві. Об'єднує

всі ресурси і процеси автотранспортного підприємства в одну зручну систему управління. Інтерфейс програми перегляду замовлень представлено на рис. 1.2.

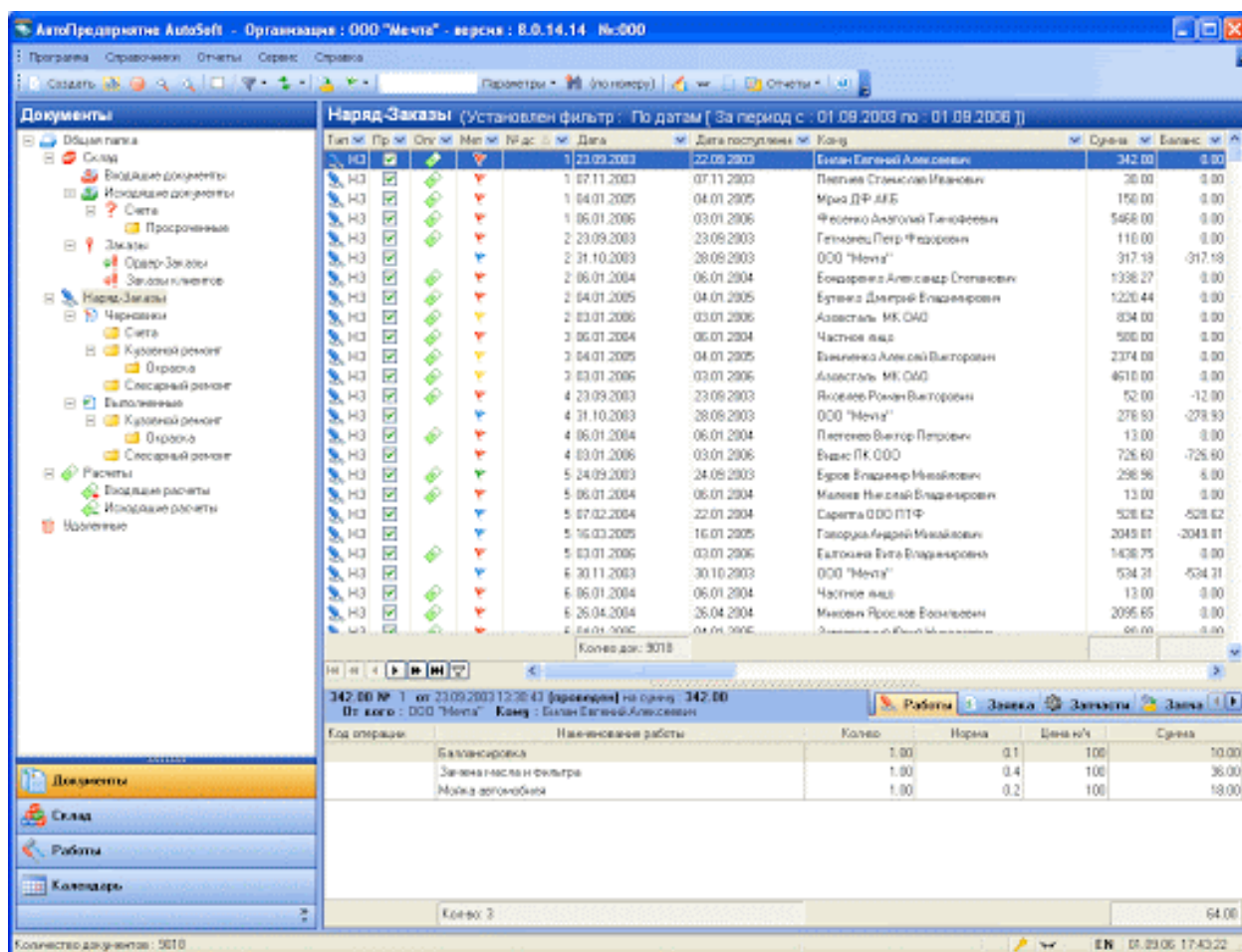


Рис. 1.2. Интерфейс програми «AutoPlan»

Програма «Контроль та облік автотранспорту». Програма призначена для обліку автомобільного транспорту і персоналу (водіїв) як в автотранспортних підприємствах, так і будь-яких підприємствах, що мають автопарки. Дозволяє контролювати своєчасну випуску полісів ОСАГО, КАСКО, ОПО і проходження Державного та міжнародного технічного огляду АТС (авто-транспортний засіб), також допомагає контролювати періодичне медичний огляд водіїв та термін дії водійського посвідчення. Інтерфейс програми представлено на рис. 1.3.

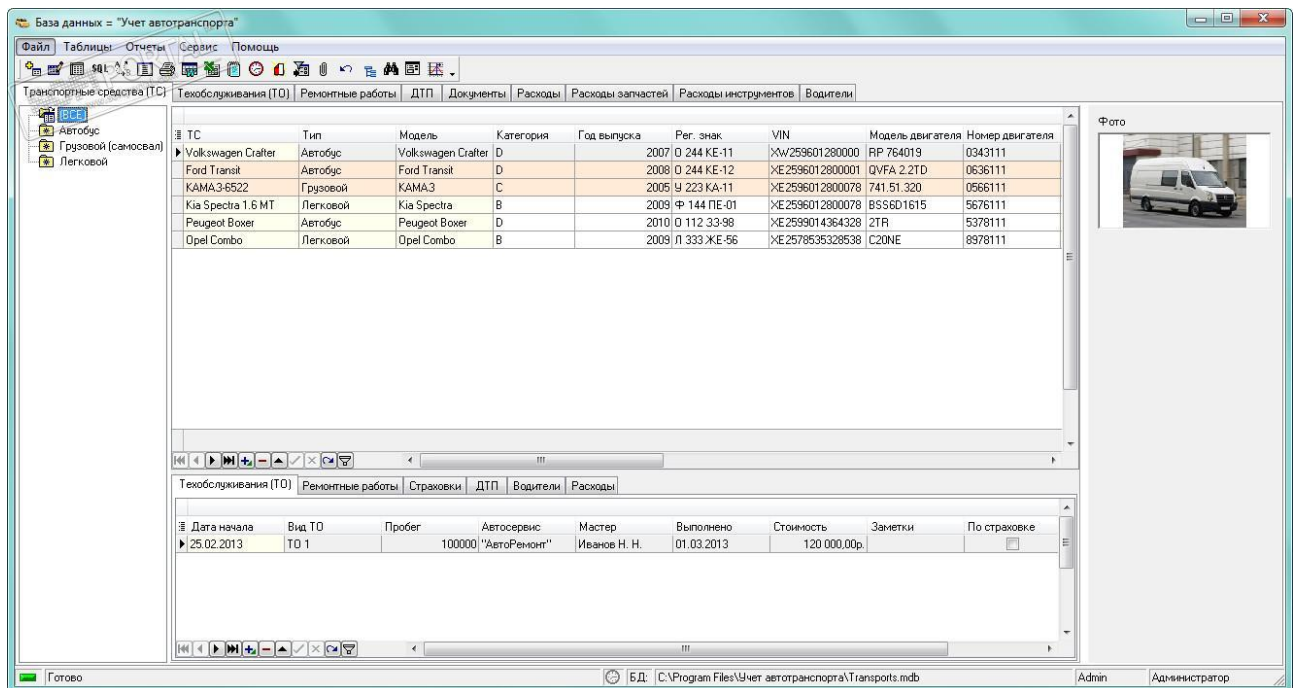


Рис. 1.3. Интерфейс программы «Контроль та облік автотранспорту»

## 1.2. Призначення розробки та область застосування

Розробка призначена полегшити та прискорити роботу диспетчера сервісу оренди автотранспорту для вантажоперевезень. Спростити формування звітності в організації за допомогою їх автоматичного формування у форматі PDF. Спростити роботу з інформацією про всіх водіїв, клієнтів, автотранспорт та укладені угоди.

## 1.3. Підстава для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу. Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;

- наказ ректора Національного технічного університету «Дніпровська політехніка» № 268-с від 18.05.2022 р;

- завдання на кваліфікаційну роботу на тему: «Розробка інформаційної автоматизованої системи для роботи диспетчера сервісу оренди автотранспорту для вантажоперевезень».

#### **1.4. Постановка завдання**

В межах данї кваліфікаційної роботи потрібно організувати базу даних, яка буде включати інформацію про контракти, клієнтів, водіїв, автомобілі та користувачів. Також потрібно розробити частину програмного додатку, яка дозволяє взаємодіяти з заданою базою даних, мати сторінки для пошуку та перегляду контрактів. Також необхідно створити інтерактивний програмний додаток із введенням даних, їх редагуванням.

Програмна система передбачає:

- зберігання даних про всі можливі автомобілі;
- зберігання даних про всіх водіїв;
- зберігання даних про всіх клієнтів;
- зберігання даних про всі договори оренди;
- сортування даних у кожній таблиці даних;
- фільтрування та пошук даних у таблицях даних;

Детальний опис вимог користувача:

1 Про водіїв потрібно зберігати таку інформацію: ім'я, адреса, номер телефону та паспортні дані.

2 Про клієнтів потрібно зберігати таку інформацію: ім'я, номер телефону та паспортні дані.

3 Про автомобілі.

4 Формування звіту за укладеними договорами за обраний проміжок часу або за весь час у форматі PDF.

5 Створення діаграми на якій показано який автомобіль було обрано найчастіше.

6 Формування системи знижок організаціям та клієнтам, що укладають договори частіше.

7 Створення системи авторизації з двома можливими рівнями доступу: «диспетчер» та «водій». Водій при вході у систему бачить лише договори, що були укладені на автомобілі з якими він працює.

8 При укладанні договору задати можливість обрати: взяти автомобіль під власний контроль, або найняти його з водієм.

9 При укладанні договору оренди диспетчером заповнюється спеціальна форма де обирається автомобіль, клієнт, водій (якщо потрібен), та детальний опис роботи.

## **1.5. Вимоги до програми або програмного виробу**

### **1.5.1. Вимоги до функціональних характеристик**

Для досягнення поставлених цілей система, що розробляється, повинна підтримувати виконання наступних функцій та дій:

Програмна система передбачає:

- зберігання даних про всі можливі автомобілі;
- зберігання даних про всіх водіїв;
- зберігання даних про всіх клієнтів;
- зберігання даних про всі договори оренди;
- сортування даних у кожній таблиці даних;
- фільтрування та пошук даних у таблицях даних;

Детальний опис вимог користувача:

- Про водіїв потрібно зберігати таку інформацію: ім'я, адреса, номер телефону та паспортні дані.

- Про клієнтів потрібно зберігати таку інформацію: ім'я, номер телефону та паспортні дані.
- Про автомобілі.
- Формування звіту за укладеними договорами за обраний проміжок часу або за весь час у форматі PDF.
- Створення діаграми на якій показано який автомобіль було обрано найчастіше.
- Формування системи знижок організаціям та клієнтам, що укладають договори частіше.
- Створення системи авторизації з двома можливими рівнями доступу: «диспетчер» та «водій». Водій при вході у систему бачить лише договори, що були укладені на автомобілі з якими він працює.
- При укладанні договору задати можливість обрати: взяти автомобіль під власний контроль, або найняти його з водієм.
- При укладанні договору оренди диспетчером заповнюється спеціальна форма де обирається автомобіль, клієнт, водій (якщо потрібен), та детальний опис роботи.

Також система повиння мати зручний та інтуїтивно зрозумілий інтерфейс.

### **1.5.2 Вимоги до інформаційної безпеки**

Для уникнення некоректної роботи програми необхідно реалізувати:

- семантичний та синтаксичний контроль вхідних даних;
- обробку виняткових ситуацій;
- виведення повідомлень про помилки;
- можливість повторного введення даних;
- можливість безперервної роботи протягом не менше 120 годин (5 діб);
- платформну незалежність.

Як складовий елемент системи, пов'язаної із торгівельною діяльністю, підсистема маршрутизації запитів повинна мати наступні характеристики:

- час відновлення після збою - 5 хв;
- час відновлення після відмови одного з елементів підсистеми не повинно перевищувати половини операційного банківського дня;
- вірогідність виникнення не більше 2 логічних помилок на 1000 операторів за 1 рік експлуатації;
- забезпечення неушкодженого стану даних, що зберігаються в базі даних, у випадку відмови підсистеми.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для нормального функціонування програми необхідно, щоб обчислювальна машина, на якій буде функціонувати веб-орієнтована підсистема, відповідала наступним вимогам:

- центральний процесор Intel Pentium, AMD;
- оперативна пам'ять 256 Мб і більше;
- об'єм пам'яті на жорсткому диску 100 Мб і більше;
- операційні системи Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10.
- доступ до мережі Internet;
- клавіатура;
- маніпулятор "миша".

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених, розроблений програмний виріб буде функціонувати відповідно до вимог щодо надійності, швидкості обробки даних і безпеки, висунутими замовником.

#### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Програма не є вимогливою до інформаційної та програмної сумісності та може завантажуватись на ПК, ноутбуках різних типів та конфігурацій під управлінням ОС Windows.



## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 2.1. Функціональне призначення системи

Розробка призначена полегшити та прискорити роботу диспетчера сервісу оренди автотранспорту для вантажоперевезень. Спростити формування звітності в організації за допомогою їх автоматичного формування у форматі PDF. Спростити роботу з інформацією про всіх водіїв, клієнтів, автотранспорт та укладені угоди.

Система, що розробляється, підтримує та виконує наступні функції та дії:

Програмна система передбачає:

- зберігання даних про всі можливі автомобілі;
- зберігання даних про всіх водіїв;
- зберігання даних про всіх клієнтів;
- зберігання даних про всі договори оренди;
- сортування даних у кожній таблиці даних;
- фільтрування та пошук даних у таблицях даних;

Детальний опис вимог користувача:

- Про водіїв потрібно зберігати таку інформацію: ім'я, адреса, номер телефону та паспортні дані.
- Про клієнтів потрібно зберігати таку інформацію: ім'я, номер телефону та паспортні дані.
- Про автомобілі.
- Формування звіту за укладеними договорами за обраний проміжок часу або за весь час у форматі PDF.
- Створення діаграми на якій показано який автомобіль було обрано найчастіше.
- Формування системи знижок організаціям та клієнтам, що укладають договори частіше.

– Створення системи авторизації з двома можливими рівнями доступу: «диспетчер» та «водій». Водій при вході у систему бачить лише договори, що були укладені на автомобілі з якими він працює.

– При укладанні договору задати можливість обрати: взяти автомобіль під власний контроль, або найняти його з водієм.

– При укладанні договору оренди диспетчером заповнюється спеціальна форма де обирається автомобіль, клієнт, водій (якщо потрібен), та детальний опис роботи.

Також система має зручний та інтуїтивно зрозумілий інтерфейс.

## **2.2. Опис застосованих математичних методів**

Оскільки особливості предметної області розв'язуваної задачі не передбачають застосування математичних методів, при розробці даного додатку математичні методи не використовувалися.

Математичні методи та функції, що використовуються системою для пошуку даних в БД, є стандартними та знаходяться у відповідних бібліотеках.

## **2.3. Опис використаних технологій та мов програмування**

Мовою написання програми обрано C#, тому що, ця мова має всі характеристики для написання об'єктно-орієнтованого проєкту, і є простою у вивченні та подальшому використанні.

C# — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (належить Microsoft).

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

Перейнявши багато від своїх попередників — мов C++, Object Pascal, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова C#, на відміну від C++, не передбачає множинне успадкування класів.

Окрім C#, середовище CLR підтримує багато інших мов програмування, зокрема в Microsoft розробляють Visual Basic .Net та F#. Крім цього існують .Net реалізації мов: Delphi.Net, IronPython, IronRuby, Boo, PascalABC.Net.

.Net дозволяє розробляти програмне забезпечення під різні операційні системи: Windows, MacOS, Linux, Android, iOS. Mono, реалізація .Net для Linux. Середовища для розробки: Visual Studio, Visual Studio Code, Unity3D, Xamarin, SharpDevelop, MonoDevelop покривають всі операційні системи і технології, та дозволяють розробляти мобільні, настільні(desktop), Web, вбудовані(embedded) додатки та сервіси.

Середовище CLR та бібліотеки класів .Net Framework включають в себе основу для багатьох технологій, які розробники можуть використовувати при створенні продуктів. Зокрема LINQ, EntityFramework, ADO.Net — технології, що використовуються для роботи з даними. Технології WindowsForms, WPF, ASP.Net, ASP.Net MVC — для створення desktop та Web-додатків.

У якості СУБД було обрано MySQL. MySQL — це реляційна система управління базами даних з відкритим вихідним кодом. В даний час ця СУБД одна з найбільш популярних в веб-додатках — переважна більшість CMS використовує саме MySQL (часто тільки її, без альтернатив), а майже всі веб-фреймворки підтримують MySQL вже на рівні базової конфігурації (без додаткових модулів).

З переваг СУБД MySQL варто відзначити простоту використання, гнучкість, низьку вартість володіння (щодо платних СУБД), а також масштабованість і продуктивність.

MySQL дозволяє зберігати цілочисельні значення зі знаком і беззнакові, довжиною в 1, 2, 3, 4 і 8 байтів, працює із строковими і текстовими даними

фіксованої і змінної довжини, дозволяє здійснювати SQL-команди SELECT, DELETE, INSERT, REPLACE і UPDATE, забезпечує повну підтримку операторів і функцій в SELECT і WHERE- частинах запитів, працює з GROUP BY і ORDER BY, підтримує групові функції COUNT (), AVG (), STD (), SUM (), MAX () і MIN (), дозволяє використовувати JOIN в запитах, в т.ч. LEFT OUTER JOIN і RIGHT OUTER JOIN, підтримує реплікацію, транзакції, роботу з зовнішніми ключами і каскадні зміни на їх основі, а також забезпечує багато інших функціональні можливості.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Є й інші типи таблиць, розроблені спільнотою.

СУБД MySQL з'явилася в 1995. Написана на C і C ++, протестована на безлічі різних компіляторів і працює на різних платформах. С 2010 року розроблення та підтримку MySQL здійснює корпорація Oracle. Продукт поширюється як під GNU GPL, так і під власною комерційною ліцензією. Однак за умовами GPL, якщо яка-небудь програма включає вихідні коди MySQL, то і ця програма теж повинна розповсюджуватися за ліцензією GPL. Для небажаючих відкривати вихідні тексти своїх програм якраз передбачена комерційна ліцензія, яка, на додаток до можливості розробки під «закритою» ліцензією, забезпечує якісну сервісну підтримку. Спільнотою розробників MySQL створені різні відгалуження — Drizzle, OurDelta, Percona Server і MariaDB, всі ці відгалуження вже існували на момент отримання прав на MySQL корпорацією Oracle.

Зараз MySQL разом з Форком MariaDB займають почесне перше місце, а слідом за ними йде PostgreSQL. Решта СУБД в веб-проектах використовуються значно рідше.

Серед розробки — WPF (Windows Presentation Foundation). Технологія WPF (Windows Presentation Foundation) є частиною екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів.

Якщо при створенні традиційних додатків на основі WinForms за отрисовку елементів управління і графіки відповідали такі частини ОС Windows, як User32 і GDI +, то додатки WPF засновані на DirectX. У цьому полягає ключова особливість рендерингу графіки в WPF: використовуючи WPF, значна частина роботи по відображенню графіки, як найпростіших кнопок, так і складних 3D-моделей, що також дозволяє скористатися апаратним прискоренням графіки.

Однією з важливих особливостей є використання мови розмітки інтерфейсу XAML, заснованою на XML: є можливість створювати насичений графічний інтерфейс, використовуючи або декларативне оголошення інтерфейсу, або код на керованих мовах C# і VB.NET, або поєднувати і те, і інше.

## **2.4. Опис структури системи та алгоритмів її функціонування**

Сценарій «Вхід в обліковий запис», представлено на рисунку 2.1. Сценарій передбачає:

1 Реєстрація та вхід до облікового запису.

1.1 Якщо користувач вперше запускає програму, то для подальшого користування йому необхідно зареєструватися у системі.

1.2 Якщо користувач вже зареєстрований у системі, то для подальшої роботи, необхідно увійти до свого облікового запису за допомогою електронної пошти вказаної при реєстрації та створеного паролю.

2 Перегляд співробітником свого облікового запису.

2.1 Перегляд власних укладених угод співробітника.

2.2 Перегляд та редагування особистої інформації.

2.3 Якщо співробітник є адміністратором, то має можливість переглядати список угод всіх співробітників, та редагувати їх особисту інформацію.

Сценарій «Введення даних про клієнтів», представлено на рисунку 2.2.

Сценарій передбачає:

1 Перегляд даних щодо всіх клієнтів у системі. Можливість відсортувати та відфільтрувати записи за бажанням.

2 Редагування даних клієнтів

3 Видалення застарілої інформації про клієнтів

Сценарій «Введення даних про співробітників», представлено на рисунку 2.3. Сценарій передбачає:

1 Перегляд даних щодо всіх співробітників у системі. Можливість відсортувати та відфільтрувати записи за бажанням.

2 Редагування даних про співробітників.

3 Видалення застарілої інформації про співробітників.



Рис. 2.1. Діаграма сценарію авторизації у системі

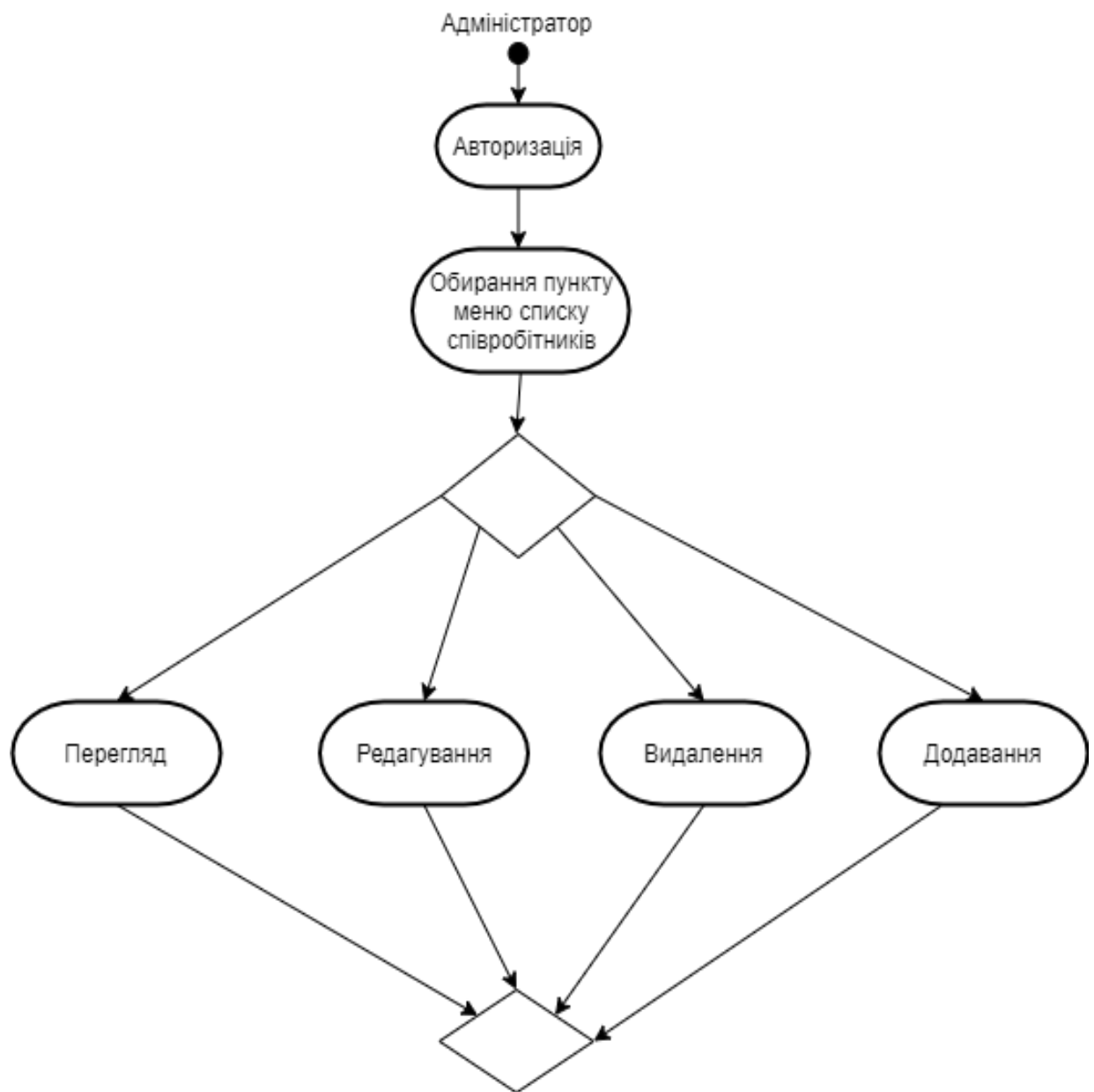


Рис. 2.2. Діаграма сценарію роботи з інформацією про співробітників

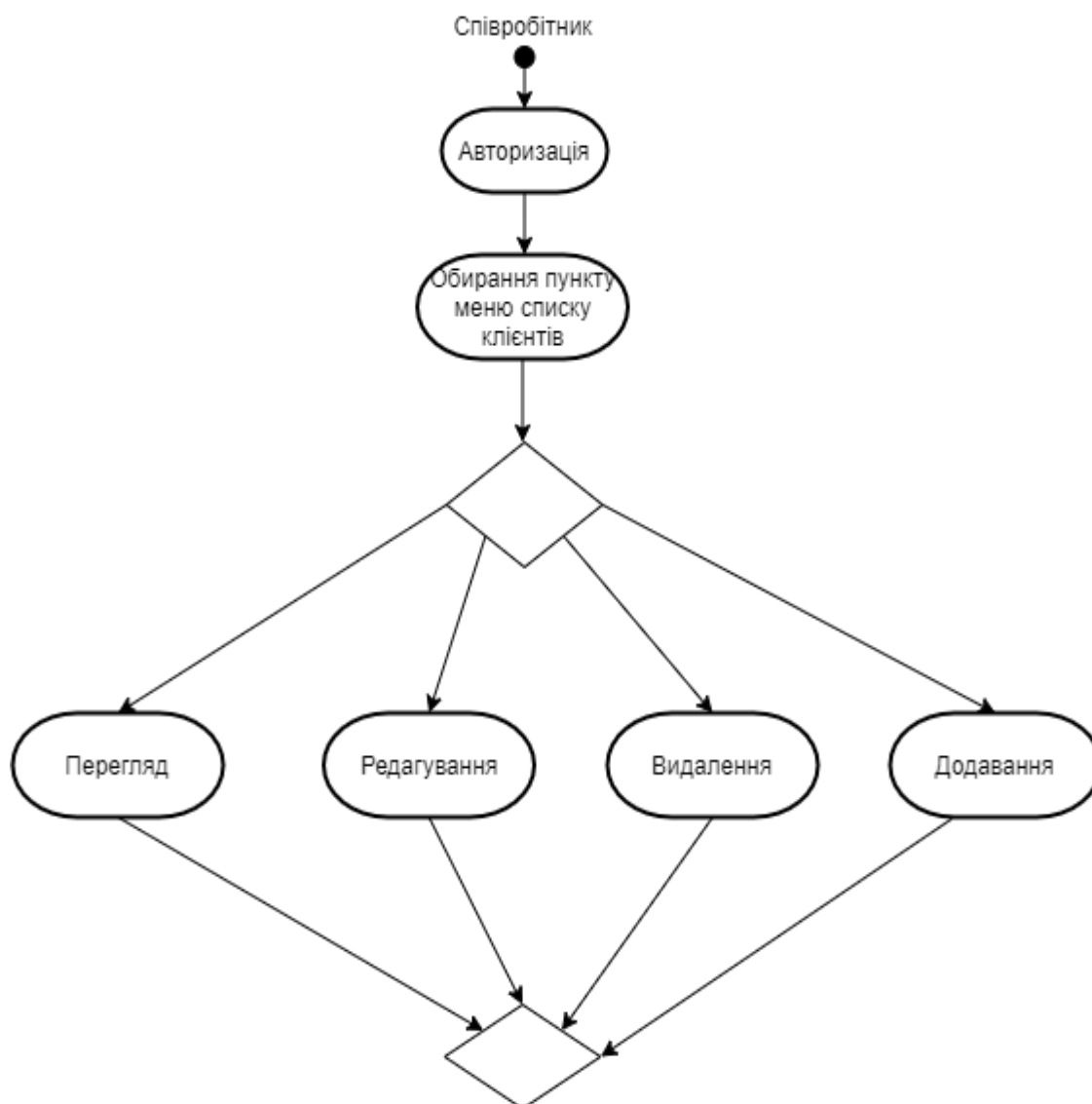


Рис. 2.3. Діаграма сценарію роботи з інформацією про клієнтів

Оскільки в базі даних зберігаються великі об'єми інформації, під час проєктування доцільно розділити базу даних на декілька сутностей (таблиць). Відношення між таблицями показані на ER-діаграмі, яка представлена на рисунку 2.4. Зміст таблиць бази даних показаний у таблицях 2.1–2.5.

Таблиця «Клієнти» зберігає таку інформацію про клієнтів: унікальний код клієнта, ім'я, прізвище, яким лицем є клієнт, мобільний телефон та електронну пошту.



Таблиця 2.1.

**Поля таблиці «Клієнти» (client)**

Назва поля	Опис	Тип	Розмір	Ключ
id	Унікальний код клієнта	Числовий	Макс. кількість символів — 8	Первинний
name	Ім'я клієнта	Символьний	Макс. кількість символів — 20	
surname	Прізвище клієнта	Символьний	Макс. кількість символів — 20	
phone	Номер телефону	Символьний	Макс. кількість символів — 15	
mail	Електронна пошта	Символьний	Макс. кількість символів — 150	

Таблиця «Водії» зберігає таку інформацію про водіїв: унікальний код водія, ім'я, прізвище, мобільний телефон та електронну пошту.

Таблиця 2.2.

**Поля таблиці «Водії» (driver)**

Назва поля	Опис	Тип	Розмір	Ключ
id	Унікальний код водія	Числовий	Макс. кількість символів — 8	Первинний
name	Ім'я водія	Символьний	Макс. кількість символів — 20	
surname	Прізвище водія	Символьний	Макс. кількість символів — 20	
phone	Номер телефону	Символьний	Макс. кількість символів — 15	
mail	Електронна пошта	Символьний	Макс. кількість символів — 150	

Таблиця «Авто» зберігає таку інформацію про автотранспорт: унікальний код транспорту, марка та модель, державний номер, максимальну завантаженість, об'єм кузова та ціну за годину оренди транспорту.

Таблиця 2.3.

**Поля таблиці «Авто» (auto)**

Назва поля	Опис	Тип	Розмір	Ключ
id	Унікальний код авто	Числовий	Макс. кількість символів — 8	Первинний
mark	Марка	Символьний	Макс. кількість символів — 20	
number	Державний номер	Символьний	Макс. кількість символів — 10	
kg	Максимальна завантаженість	Числовий	Макс. кількість символів — 10	
volume	Об'єм	Числовий	Макс. кількість символів — 15	
price	Ціна аренди	Числовий	Макс. кількість символів — 10	

Таблиця «Користувач» зберігає таку інформацію про користувачів: унікальний код користувача, ім'я, прізвище, мобільний телефон, логін та пароль.

Таблиця 2.4.

**Поля таблиці «Користувач» (user)**

Назва поля	Опис	Тип	Розмір	Ключ
id	Унікальний код користувача	Числовий	Макс. кількість символів — 8	Первинний
name	Ім'я користувача	Числовий	Макс. кількість символів — 30	
surname	Прізвище користувача	Символьний	Макс. кількість символів — 30	
phone	Номер телефону	Числовий	Макс. кількість символів — 15	
login	Логін	Символьний	Макс. кількість символів — 32	
pass	Пароль	Символьний	Макс. кількість символів — 32	

Таблиця «Договори» зберігає таку інформацію про договори: унікальний код, ціна, дата укладання, дата виконання, унікальний код користувача,

унікальний код водія, унікальний код, клієнта, унікальний код автотранспорту та статус виконання.

Таблиця 2.5.

**Поля таблиці «Договору» (orders)**

Назва поля	Опис	Тип	Розмір	Ключ
id	Унікальний номер договору	Числовий	Макс. кількість символів — 8	Первинний
prise	Вартість договору	Числовий	Макс. кількість символів — 10	
userid	Номер користувача	Числовий	Макс. кількість символів — 8	Зовнішній
autoid	Номер автомобіля	Числовий	Макс. кількість символів — 8	Зовнішній
clientid	Номер клієнта	Числовий	Макс. кількість символів — 8	Зовнішній
driverid	Номер водія	Числовий	Макс. кількість символів — 8	Зовнішній
datestart	Дата укладення договору	Дата		
dateend	Дата кінця договору	Дата		
state	Статус виконання	Символьний	Макс. кількість символів — 15	

ER-діаграма зв'язку таблиць бази даних наведена нижче на рис. 2.4.

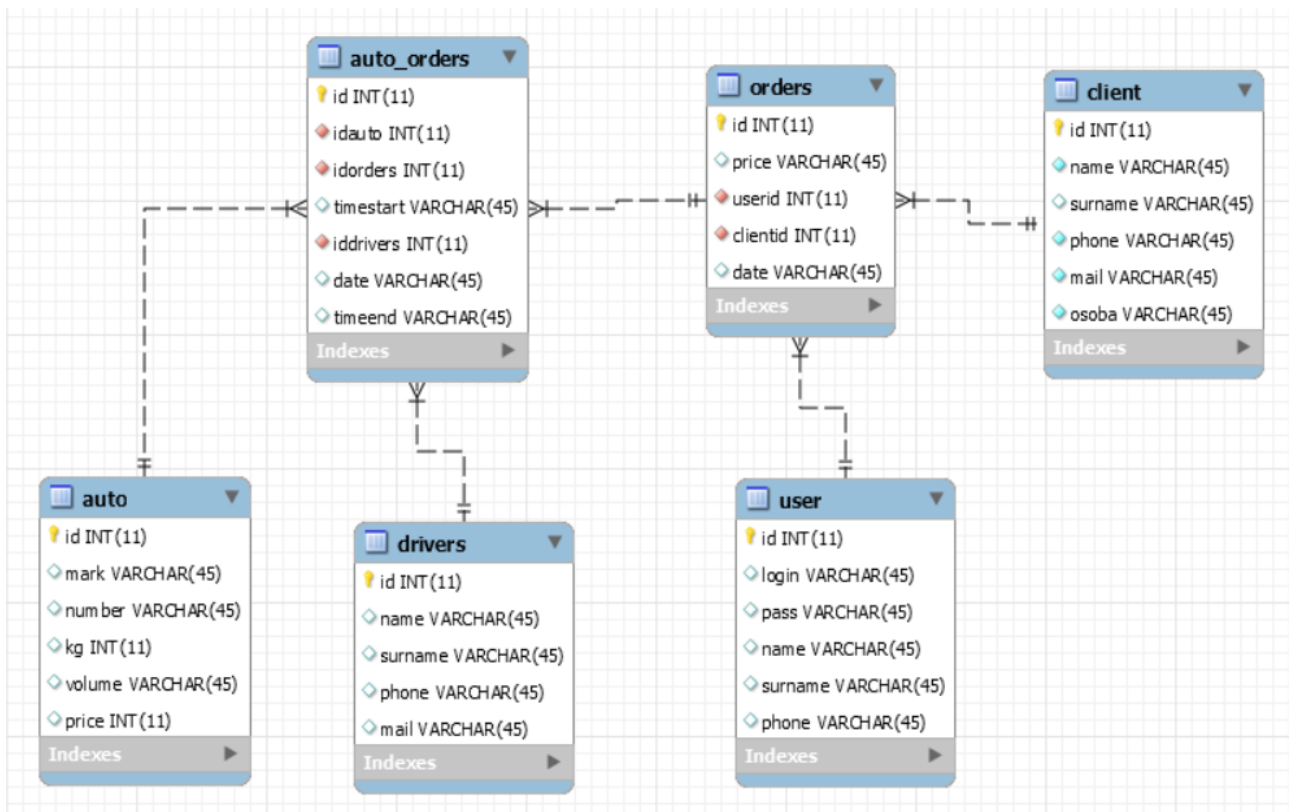


Рис. 2.4. ER-діаграма зв'язку таблиць бази даних

Інтерфейс програмного додатку складається з форм WPF на яких розташовуються різні компоненти за допомогою яких користувач працює з програмою. Кожен елемент має набір властивостей, що задають його ім'я, тип, стиль та певні обмеження на введені дані.

Інтерфейс програмного додатку даного дипломного проекту складається з форм, кожна з яких має власні компоненти та обробники подій:

1 LoginForm — форма авторизації користувача, перша форма що відкривається при запуску програми. Форма має такі обробники подій:

- LoginButton\_Click() — створення запиту щодо авторизації до бази даних та перевірка на коректність введених даних;
- RegOpenButton\_Click() — відкриття форми реєстрації з форми авторизації;
- LoginForm\_Closed() — закриває програму при закритті форми авторизації.

2 RegForm — форма реєстрації нового користувача, відкривається з форми авторизації. Форма має такі обробники подій:

- RegForm\_Load() — ініціалізує поля заповнення даних користувача при реєстрації;
- IsUserExists() — перевірка на вже існуючих користувачів при реєстрації;
- regButton\_click() — додавання нового користувача до бази даних;
- backButton\_click() — повернення користувача.

3 MainForm — головна форма програми відкривається після успішної авторизації користувача. Форма має такі обробники подій:

- MainForm\_Load() — заповнює таблиці програми відповідними таблицями з бази даних;
- Exit\_Form() — вихід з головної форми програми на вікно авторизації;
- AC\_Form() — відкриття форми додавання нового клієнта;
- AD\_Form() — відкриття форми додавання нового водія;
- AA\_Form() — відкриття форми додавання нового автотранспорту;
- AO\_Form() — відкриття форми додавання нового контракту;
- RC\_Form() — відкриття форми редагування існуючого клієнта;
- RD\_Form() — відкриття форми редагування існуючого водія;
- RA\_Form() — відкриття форми редагування існуючого автотранспорту;
- RO\_Form() — відкриття форми редагування існуючого контракту
- Del\_cl() — видалення запису про клієнта;
- Del\_dr() — видалення запису про водія;
- Del\_au() — видалення запису про автотранспорт;
- Del\_or() — видалення запису про контракт
- Search\_cl() — пошук записів про клієнтів;
- Search\_dr() — пошук записів про водіїв;
- Search\_au() — пошук записів про автотранспорт;
- Search\_or() — пошук записів про контракти;

- Stat\_create() — створення статистики та її візуального відображення за обраної таблицею даних;

- Abprog\_op() — відкриття форми «Про Додаток»;

- Profile\_open() — відкриття форми профіля користувача.

4 AddClientForm — форма додавання нового клієнта, відкривається з головної форми програми. Форма має такий обробник події:

- Addclient\_click() — додає нового клієнта до системи.

– AddDriverForm — форма додавання нового водія, відкривається з головної форми програми. Форма має такий обробник події:

- Adddriver\_click() — додає нового водія до системи.

5 AddTransportForm — форма додавання нового автотранспорту, відкривається з головної форми програми. Форма має такі обробники подій:

- Addtrans\_click() — додає новий автотранспорт до системи;

- Photo\_click() — змінює фото автотранспорту.

6 AddOrderForm — форма додавання нового контракту відкривається з головної форми програми. Форма має такий обробник події:

- AddOrder\_click() — додає новий контракт до системи.

7 RedClientForm — форма редагування запису про клієнта, відкривається з головної форми програми. Форма має такий обробник події:

- Redclient\_click() — редагує інформацію про клієнта у системі.

8 RedDriverForm — форма редагування запису про водія, відкривається з головної форми програми. Форма має такий обробник події:

- Reddriver\_click() — редагує інформацію про водія у системі.

9 RedTransportForm — форма редагування запису про автотранспорт, відкривається з головної форми програми. Форма має такі обробники подій:

- Redtrans\_click() — редагує інформацію про автотранспорт у системі;

- Photored\_click() — змінює фото автотранспорту при редагуванні.

10 RedOrderForm — форма редагування записів про контракти відкривається з головної форми програми. Форма має такий обробник події:

– `redOrder_click()` — редагує інформацію про існуючий контракт у системі.

11 `ProfileForm` — форма перегляду особистого профілю користувача. Форма має такі обробники подій:

- `profile_red()` — редагує інформацію особистого профіля користувача;
- `Photored_profile()` — змінює фото в особистому профілі користувача.

12 `AboutProgForm` — форма з коротким описом програми розділу «Про Додаток».

## **2.5. Обґрунтування та організація вхідних та вихідних даних програми**

В якості вхідних даних системна використовує інформацію запиту від користувача та інформація БД.

В якості вхідних даних використовуються заповнені форми та звіти системи.

## **2.6. Опис роботи розробленої системи**

### **2.6.1. Використані технічні засоби**

Програма не є вимогливою до складу та параметрів технічних засобів та може завантажуватись на ПК, ноутбуках різних типів та конфігурацій під управлінням різних ОС Windows.

Систему розроблено на ноутбуці Ноутбук Dell Inspiron G5 5500:

Екран 15.6" WVA (1920x1080) Full HD, глянсовий з антивідблисковим покриттям / Intel Core i5-10300H (2.5 — 4.5 ГГц) / RAM 8 ГБ / SSD 1 ТБ / nVidia GeForce GTX 1650 Ti, 4 ГБ / без ОД / LAN / Wi-Fi / Bluetooth / вебкамера

### **2.6.2. Використані програмні засоби**

Програму створено в середовищі програмування Microsoft Visual Studio 2019 мовою C#.

Для роботи з СУБД використовувалися phpMyAdmin — веб-додаток з відкритим кодом, написаний мовою PHP і є веб-інтерфейсом для адміністрування СУБД MySQL. PhpMyAdmin дозволяє через браузер і не тільки здійснювати адміністрування сервера MySQL, запускати команди SQL та переглядати вміст таблиць та баз даних.

Для пошуку інформації використано Google Chrome.

### **2.6.3. Виклик та завантаження програми**

Спосіб виклику програми з відповідного носія даних та умови його завантаження є стандартними для запуску виконуючих файлів при роботі в ОС Windows. Додаткових чи специфічних вимог щодо запуску програми не встановлено, програма не потребує спеціального завантаження та налаштування і може бути скопійована на інші носії інформації.

Для роботи потрібен ПК чи наутбук з ОС Windows. Щоб установити програму на комп'ютер, треба скопіювати папку «Program» та відкрити файл .exe.

### **2.6.4. Опис інтерфейсу користувача**

Після запуску програми відкривається вікно авторизації, як представлено на рис. 2.5.



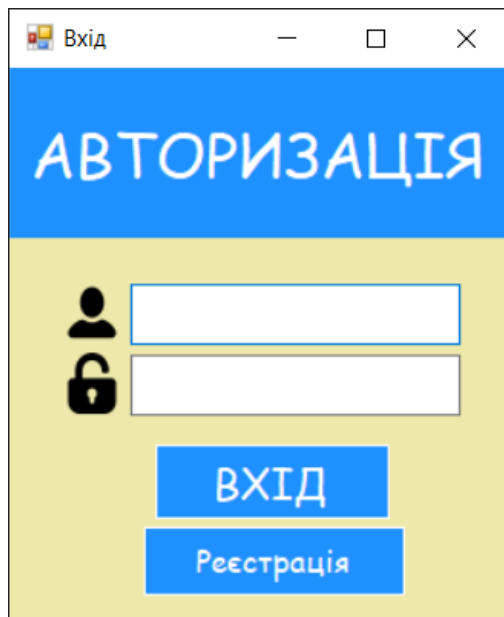


Рис. 2.5. Форма авторизації

У поля потрібно ввести дані для входу: пароль та логін що були вказані при реєстрації а натиснути кнопку «Вхід». Якщо користувач ще не зареєстрований у системі, то потрібно натиснути кнопку «Реєстрація». Після цього відкриється форма реєстрації, представлено на рис. 2.6.

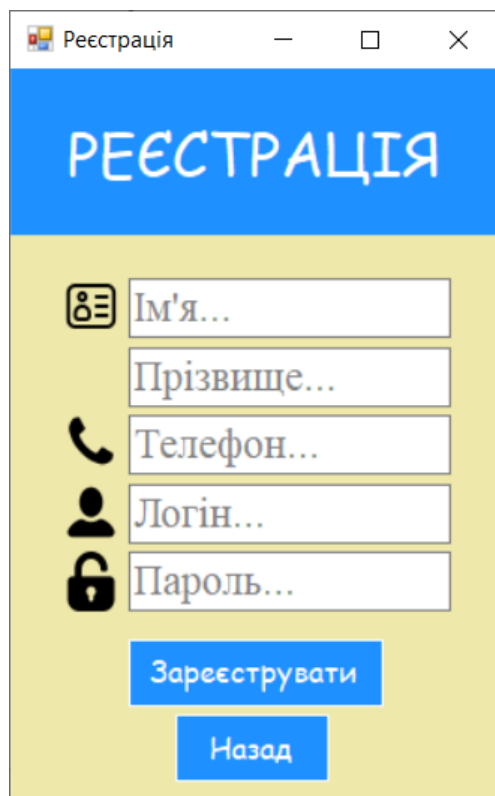


Рис. 2.6. Форма реєстрації у системі

У відповідні поля користувач повинен ввести ім'я, прізвище, телефон, логін та пароль. Після цього натиснути кнопку «Зареєструвати». При порожніх полях програма виводить повідомлення про помилку, представлено на рис. 2.7.

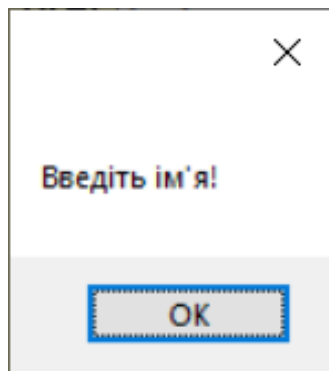


Рис. 2.7. Повідомлення про помилку

Після реєстрації користувача програма повертається на форму авторизації та пропонує ввести дані ще раз. Якщо користувач введе невірні дані, програма виводить помилку, представлено на рис. 2.8.

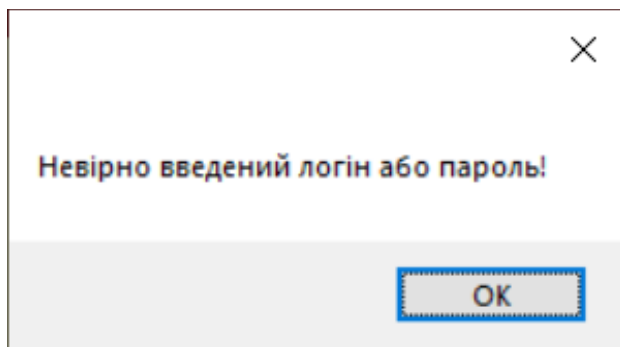


Рис. 2.8. Повідомлення про помилку

Після авторизації користувачу відкривається головна форма програми, представлено на рисунку 2.9.

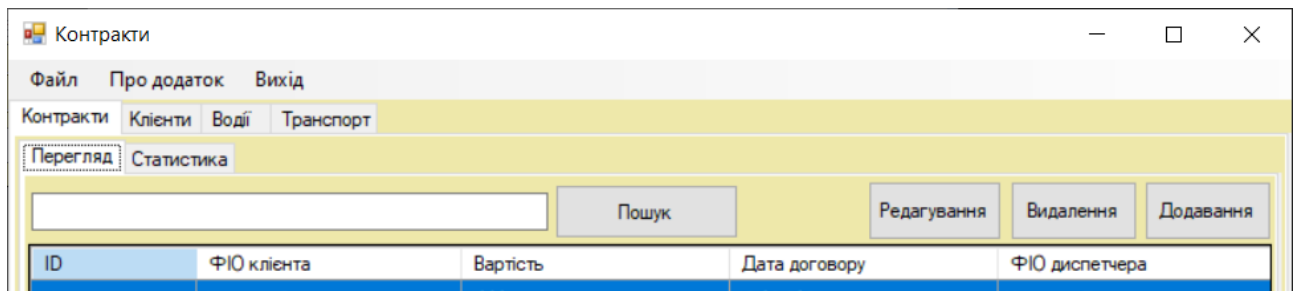


Рис. 2.9. Головна форма програми

На формі знаходяться вкладки з таблицями даних: клієнти, водії, контракти та авто. Для додавання нового клієнту потрібно перейти на вкладку «Клієнти» та натиснути кнопку «Додавання». Відкриється форма додавання клієнта, зображено на рис. 2.10.

Рис. 2.10. Форма додавання клієнта

Для видалення клієнта потрібно обрати його з таблиці та натиснути кнопку «Видалення». Для редагування інформації про клієнта потрібно обрати його з таблиці та натиснути кнопку «Редагування». Відкриється форма редагування, зображено на рис. 2.11.

Редагування клієнта

# КЛІЄНТ

Особа: Фізична

Ім'я: Алексей

Прізвище: Рев

Контактний номер: +38023246

e-mail: agd@gmail.com

Редагувати

Рис. 2.11. Форма редагування клієнта

Для пошуку серед клієнтів потрібно ввести ім'я у полі над таблицею та натиснути кнопку «Пошук», представлено на рис. 2.12.

Пошук

Рис. 2.12. Пошук серед клієнтів

Для додавання нового водія потрібно перейти на вкладку «Водії» та натиснути кнопку «Додавання». Відкриється форма додавання водія, зображено на рис. 2.13.

Додавання нового водія

# ВОДІЙ

Ім'я:

Прізвище:

Контактний номер:

e-mail:

Додати

Рис. 2.13. Форма додавання водія

Для видалення водія потрібно обрати його з таблиці та натиснути кнопку «Видалення». Для редагування інформації про водія потрібно обрати його з таблиці та натиснути кнопку «Редагування». Відкриється форма редагування, зображено на рис. 2.14.

Редагування водія

# ВОДІЙ

Ім'я:

Прізвище:

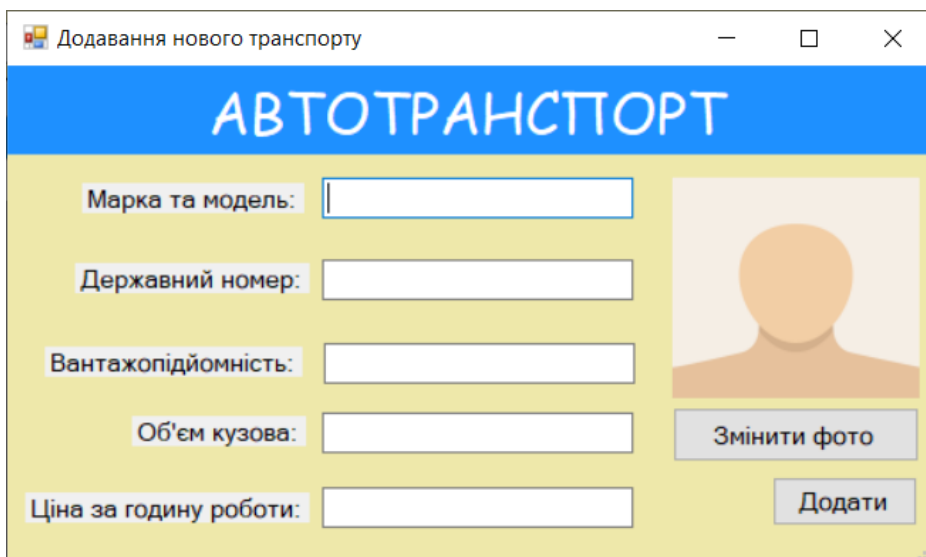
Контактний номер:

e-mail:

Редагувати

Рис. 2.14. Форма редагування водія

Для додавання нового авто потрібно перейти на вкладку «Транспорт» та натиснути кнопку «Додавання». Відкриється форма додавання авто, зображено на рис. 2.15.



Додавання нового транспорту

## АВТОТРАНСПОРТ

Марка та модель:

Державний номер:

Вантажопідйомність:

Об'єм кузова:

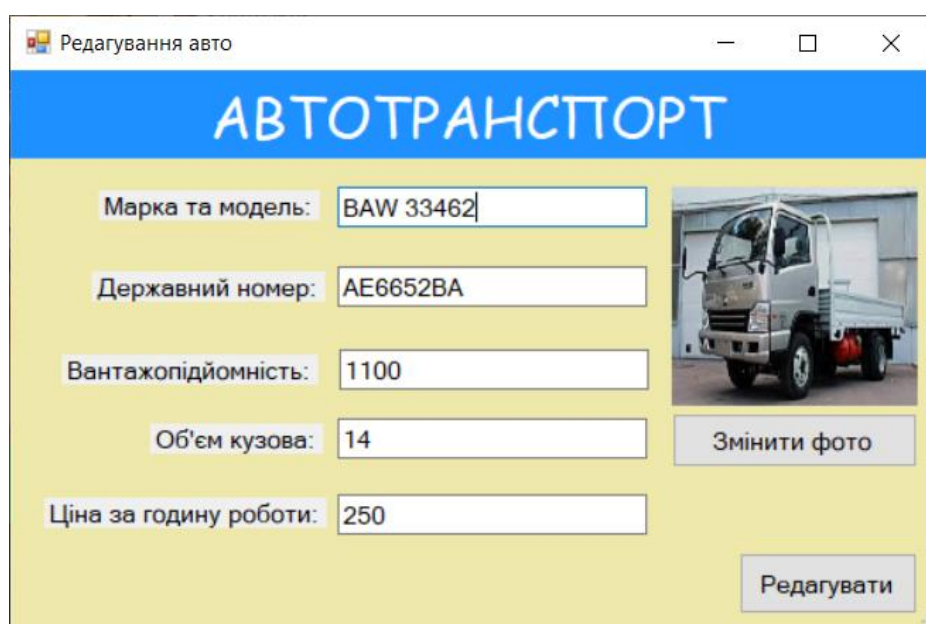
Ціна за годину роботи:

Змінити фото

Додати

Рис. 2.15. Форма додавання автотранспорту

Для видалення водія потрібно обрати його з таблиці та натиснути кнопку «Видалення». Для редагування інформації про водія потрібно обрати його з таблиці та натиснути кнопку «Редагування». Відкриється форма редагування, зображено на рис. 2.16.



Редагування авто

## АВТОТРАНСПОРТ

Марка та модель: BAW 33462

Державний номер: AE6652BA

Вантажопідйомність: 1100

Об'єм кузова: 14

Ціна за годину роботи: 250

Змінити фото

Редагувати

Рис. 2.16. Форма редагування автотранспорту

Для додавання нового контракту потрібно перейти на вкладку «Контракти» та натиснути кнопку «Додавання». Відкриється форма додавання контракту, зображено на рис. 2.17.

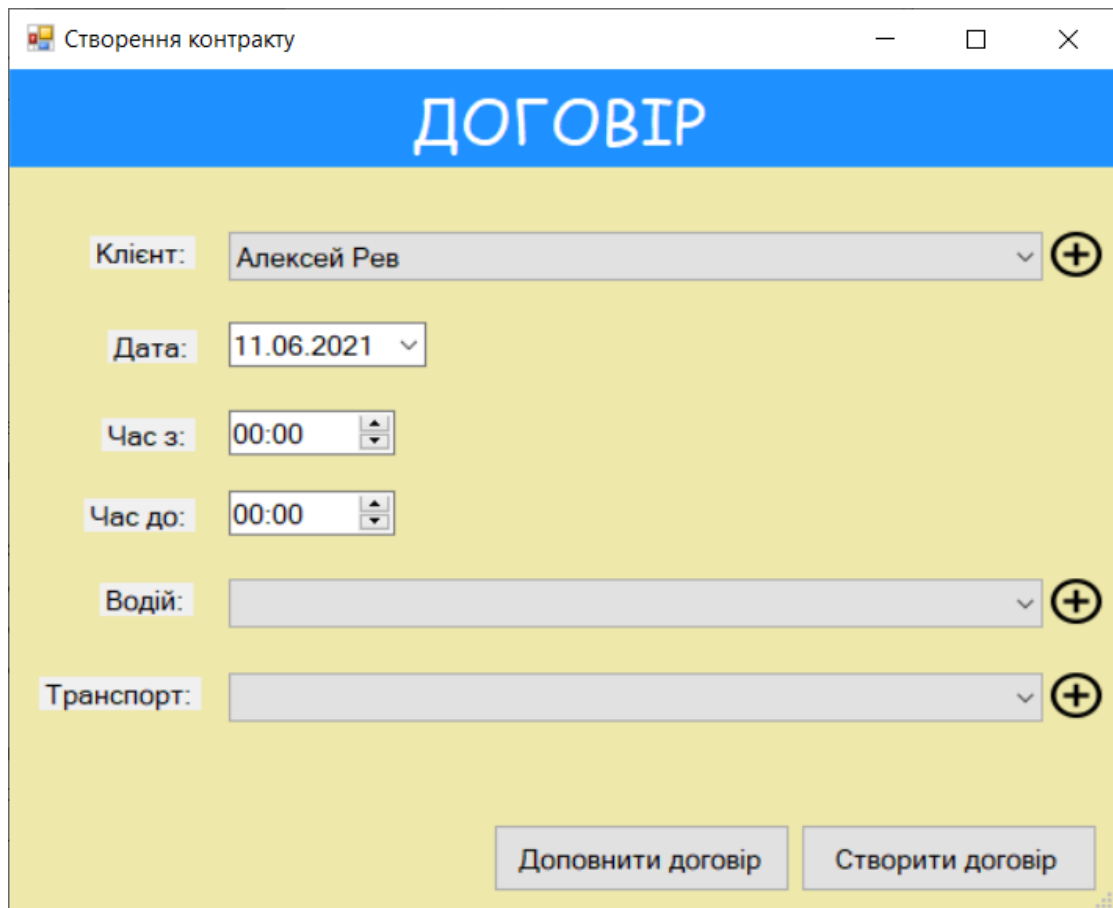


Рис. 2.17. Форма додавання контракту

Для видалення клієнта потрібно обрати його з таблиці та натиснути кнопку «Видалення». Для редагування інформації про клієнта потрібно обрати його з таблиці та натиснути кнопку «Редагування». Відкриється форма редагування.

Для перегляду статистики та діаграми по кожній таблиці даних перейдіть на вкладку «Статистика» кожного розділу, зображено на рис. 2.18.

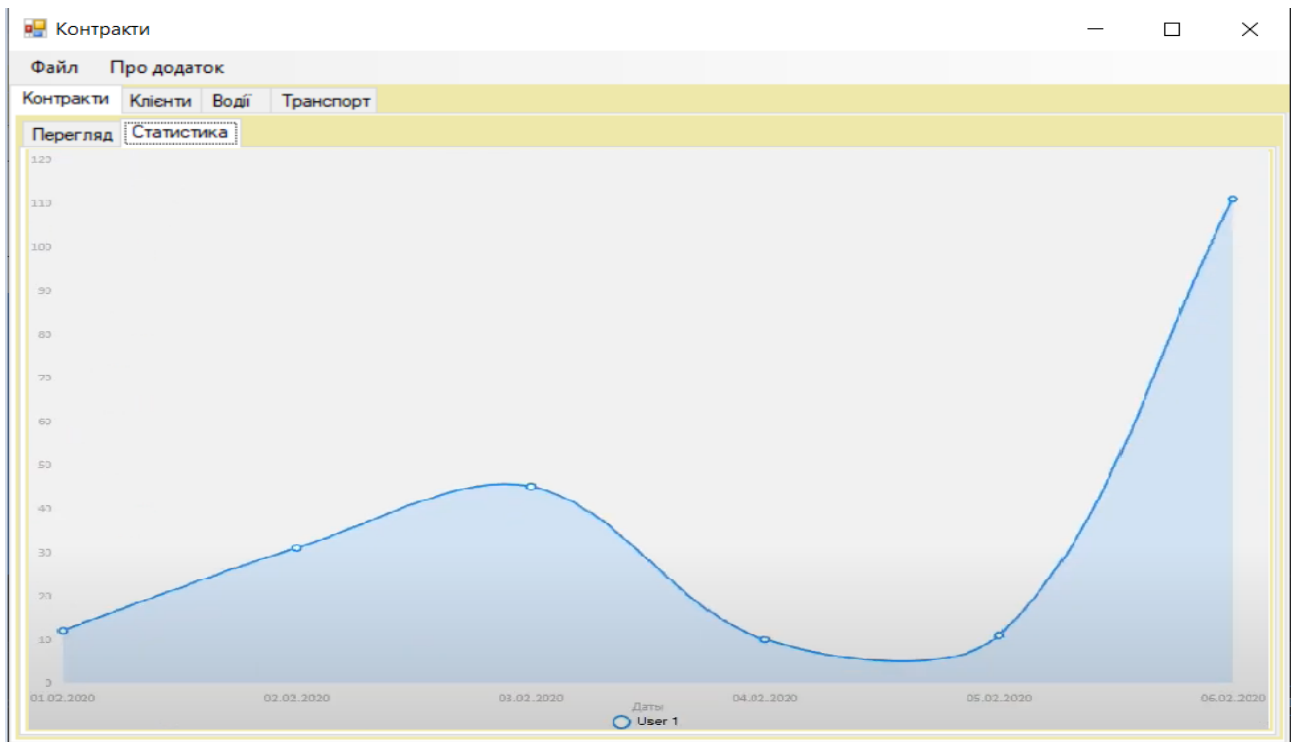


Рис. 2.18. Статистика по договорам за датую

При двійному кліку на обраний договір відкривається додаткова інформація про нього, представлено на рис. 2.19.

ID	ФІО водія	Модель	Державний номер	Дата та час роботи
43	Олег Игн	BAW 33462	AE6652BA	14.06.2021 16:00 - 20:00
43	Арсений III	ГАЗ 2705-216	AE6722AH	14.06.2021 16:00 - 20:00

Рис. 2.19. Форма додаткової інформації про договір



При натисканні кнопки «Сформувати договір» формується документ з основною інформацією.

## РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

### 3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1200;
2. коефіцієнт складності програми – 1,6;
3. коефіцієнт корекції програми в ході її розробки – 0,05;
4. годинна заробітна плата програміста – 120 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
7. вартість машино-години ЕОМ – 13 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_{\partial}, \text{ людино-годин, (3.1)}$$

де  $t_o$  - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

$t_u$  - витрати праці на дослідження алгоритму рішення задачі;

$t_a$  - витрати праці на розробку блок-схеми алгоритму;

$t_n$  - витрати праці на програмування по готовій блок-схемі;

$t_{отл}$  - витрати праці на налагодження програми на ЕОМ;

$t_{\partial}$  - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p),$$

де  $q$  - передбачуване число операторів (1200);

$C$  - коефіцієнт складності програми (1,6);

$p$  - коефіцієнт корекції програми в ході її розробки (0,05).

Звідси умовне число операторів в програмі:

$$Q = 1,6 \cdot 1200 \cdot (1 + 0,05) = 2016$$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k}, \text{ людино-годин,}$$

де  $B$  - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

$k$  - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає 1,2.

Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ( $B = 1,2$ ). З урахуванням коефіцієнта кваліфікації  $k = 1,2$ , отримуємо витрати праці на вивчення опису завдання:

$$t_u = (2016 \cdot 1,2) / (75 \cdot 1,2) = 26,88 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин, (3.2)}$$

де  $Q$  – умовне число операторів програми;

$k$  – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.2), отримаємо:

$$t_a = 2016 / (20 \cdot 1,2) = 84 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин.}$$

$$t_n = 2016 / (25 \cdot 1,2) = 67,2 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин.}$$

$$t_{oml} = 2016 / (5 \cdot 1,2) = 336 \text{ людино-годин.}$$

- за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин.}$$

$$t_{oml}^k = 1,5 \cdot 336 = 504 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин,}$$

де  $t_{\partial p}$  - трудомісткість підготовки матеріалів і рукопису:

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин,}$$

$t_{\partial o}$  - трудомісткість редагування, печатки й оформлення документації:

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ людино-годин.}$$

Підставляючи відповідні значення, отримаємо:

$$t_{\partial p} = 2016 / (18 \cdot 1,2) = 93,33 \text{ людино-годин.}$$

$$t_{\partial o} = 0,75 \cdot 93,33 = 70 \text{ людино-годин.}$$

$$t_{\partial} = 93,33 + 70 = 163,33 \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 26,88 + 84 + 67,2 + 336 + 163,33 = 727,41 \text{ людино-годин.}$$

### 3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ  $K_{ПО}$  включають витрати на заробітну плату виконавця програми  $Z_{ЗП}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн,}$$

де:  $t$  - загальна трудомісткість, людино-годин;

$C_{ПР}$  - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 120 грн / год, отримуємо:

$$Z_{ЗП} = 727,41 \cdot 120 = 87\,289,2 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн, (3.3)}$$

де  $t_{отл}$  - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$  - вартість машино-години ЕОМ, грн/год (13 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{mv} = 336 \cdot 13 = 4368 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 87\,289,2 + 4368 = 91\,657,2 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ міс.}$$

де  $B_k$  - число виконавців (дорівнює 1);

$F_p$  - місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p=176$  годин).

Звідси витрати на створення програмного продукту:

$$T = 727,41 / 1 \cdot 176 \approx 4 \text{ міс.}$$

**Висновок.** Програмне забезпечення призначене полегшити та прискорити роботу диспетчера сервісу оренди автотранспорту для вантажоперевезень. Спростити формування звітності в організації за допомогою їх автоматичного формування у форматі PDF. Спростити роботу з інформацією про всіх водіїв, клієнтів, автотранспорт та укладені угоди. Вартість даного програмного забезпечення становить 91 657,2. грн. і не вимагає додаткових витрат як при розробці програми. Очікуваний час розробки становить 4 місяці. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено програмний додаток, що допомагає автоматизувати робоче місце диспетчеру автопарку.

Розроблений додаток має інтуїтивно зрозумілий графічний інтерфейс і користувач не потребує спеціальних знань для роботи з ним.

Таким чином, підводячи підсумки дипломного проектування, слід зазначити, що в під час роботи був розроблений програмний додаток мовою С#, що виконує роботу з базою даних, формує звітність та статистику. Програма переглядає, додає, редагує та видаляє інформацію з бази даних. Розроблений додаток виконує всі поставлені завдання та задовольняє вимоги клієнта.

Всі поставлені в роботі завдання виконано в повному обсязі.

Також у кваліфікаційній роботі було визначено трудомісткість розробленої системи (727 люд-год), проведений підрахунок вартості роботи по створенню програми (91 657,2 грн) та розраховано час на її створення (4 міс).



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Р. Грофф, Пол Н. Вайнберг, Ендрю Дж. Оппель SQL. Повне керівництво Джеймс: Видавництво "Вільямс", 2015.-960стор. іл.
2. Володимир Гайдаржі, Ігор Ізварін "Базі даних в інформаційних системах": Видавництво Університет; Україна, 2018.– 418с. Іл.
3. Аллен Дж. Тейлор Видавництво Діалектика, Вільямс "SQL для чайників": 2014. - 416с.
4. Ентоні Молінаро "SQL. Збірник рецептів": Перекладач Н. Шатохіна, Видавництво "Символ-Плюс", 2009. – 672 стор.
5. Альфред В. Ахо. Компілятори Принципи, технології та інструментарій / Альфред В. Ахо та ін., 2015. – 266 с.
6. Бішоп Дж. С # у короткому викладі / Дж. Бішоп, Н. Хорспул., 2013. – 472 с.
7. Вагнер Білл. С# Ефективне програмування / Білл Вагнер, 2013. – 320с.
8. Зіборов Віктор. Visual С# 2010 на прикладах / Віктор Зіборов., 2011. - 432 с.
9. Ішкова Е. А. Самовчитель С #. Початки програмування/Е.А. Ішкова., 2013. – 496 с.
10. Лотка Рокфорд. С# та CSLA .NET Framework. Розробка бізнес-об'єктів / Рокфорд Лотка., 2010. - 816 с.
11. Мак-Дональд Метью. Silverlight 5 з прикладами на С# для професіоналів / Метью МакДональд., 2013. - 848 с.
12. Подбельський В. В. Мова С #. Базовий курс/В.В. Ірпінь., 2011. – 384с.
13. Ріхтер Джеффри. CLR за допомогою С#. Програмування на платформі Microsoft .NET Framework 4.0 мовою С# / Джеффри Ріхтер., 2013. – 928 с.
14. Троелсен Ендрю. Мова програмування С# 5.0 та платформа .NET 4.5 / Ендрю Троелсен., 2015. - 486 с.

## КОД ПРОГРАМИ

## Лістинг 1 – AddClientForm

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

namespace Diplom
{
    public partial class AddClientForm : Form
    {
        public AddClientForm()
        {
            InitializeComponent();
        }

        private void AddClientForm_Load(object sender, EventArgs e)
        {
            comboBox1.SelectedIndex = 0;
        }

        private void comboBox1_SelectedIndexChanged(object sender,
        EventArgs e)
        {
            if (comboBox1.SelectedIndex == 1)
            {
                label2.Visible = false;
                textBox2.Visible = false;
                label11.Visible = false;
                label6.Visible = true;
                textBox2.Text = "-----";
            }
            else
            {
                label2.Visible = true;
                textBox2.Visible = true;
                label6.Visible = false;
                label11.Visible = true;
                textBox2.Clear();
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
        }
    }
}
```

```

        MySqlCommand command = new MySqlCommand("insert into client
(name, surname, phone, mail, osoba) values (@n, @s, @p, @m, @o)",
db.getConnection());
        command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
textBox1.Text;
        command.Parameters.Add("@s", MySqlDbType.VarChar).Value =
textBox2.Text;
        command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
textBox3.Text;
        command.Parameters.Add("@m", MySqlDbType.VarChar).Value =
textBox4.Text;
        command.Parameters.Add("@o", MySqlDbType.VarChar).Value =
comboBox1.Text;
        db.openConnection();
        if (command.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Новий користувач додан!");
        }
        else
            MessageBox.Show("Не вдалося додати нового
користувача!");
        db.closeConnection();
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Text="+380";
        textBox4.Clear();
    }
}
}

```

## Лістинг 2 – AddDriverForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class AddDriverForm : Form
    {
        public AddDriverForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
            MySqlCommand command = new MySqlCommand("insert into drivers
(name, surname, phone, mail) values (@n, @s, @p, @m)",
db.getConnection());

```

```

        command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
textBox1.Text;
        command.Parameters.Add("@s", MySqlDbType.VarChar).Value =
textBox2.Text;
        command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
textBox3.Text;
        command.Parameters.Add("@m", MySqlDbType.VarChar).Value =
textBox4.Text;
        db.openConnection();
        if (command.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Новий водій додан!");
        }
        else
            MessageBox.Show("Не вдалося додати нового водія!");
        db.closeConnection();
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Text = "+380";
        textBox4.Clear();
    }
}
}

```

### Лістинг 3 – AddOrdersForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class AddOrdersForm : Form
    {
        public AddOrdersForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
            string date= DateTime.Now.ToShortDateString();
            MySqlCommand command = new MySqlCommand("insert into orders
(userid, clientid, date) values (@ui, @ci, @date)", db.getConnection());
            command.Parameters.Add("@ci", MySqlDbType.Int32).Value =
Convert.ToInt32(comboBox1.SelectedValue);
            command.Parameters.Add("@ui", MySqlDbType.Int32).Value =
Stat.u_id;
            command.Parameters.Add("@date", MySqlDbType.String).Value =
date;

```

```

        db.openConnection();
        if (command.ExecuteNonQuery() == 1)
        {
            DataTable table = new DataTable();
            MySqlDataAdapter adapter = new MySqlDataAdapter("select
max(id) from orders;", db.getConnection());
            adapter.Fill(table);
            int orders_id = Convert.ToInt32(table.Rows[0][0]);
            MySqlCommand command2 = new MySqlCommand("insert into
auto_orders (idorders, idauto, iddrivers, date, timestart, timeend)
values (@ido ,@ida, @idd, @dt, @ts, @te)", db.getConnection());
            command2.Parameters.Add("@idd", MySqlDbType.Int32).Value
= Convert.ToInt32(comboBox2.SelectedValue);
            command2.Parameters.Add("@ida", MySqlDbType.Int32).Value
= Convert.ToInt32(comboBox3.SelectedValue);
            command2.Parameters.Add("@ido", MySqlDbType.Int32).Value
= orders_id;
            command2.Parameters.Add("@ts", MySqlDbType.String).Value
= dateTimePicker2.Value.ToString("HH:mm");
            command2.Parameters.Add("@te", MySqlDbType.String).Value
= dateTimePicker3.Value.ToString("HH:mm");
            command2.Parameters.Add("@dt", MySqlDbType.String).Value
= dateTimePicker1.Value.ToShortDateString();
            if (command2.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Новий контракт створено!");
            }
            else
                MessageBox.Show("Не вдалося додати новий запис до
контракту!");
        }
        else
            MessageBox.Show("Не вдалося створити новий контракт!");
        db.closeConnection();
        combobox_date();
    }

    private void dateTimePicker1_ValueChanged(object sender,
EventArgs e)
    {
        combobox_date();
    }

    private void combobox_date()
    {
        comboBox2.Enabled = true;
        comboBox3.Enabled = true;
        DB db = new DB();
        DataTable dataSet_dr = new DataTable();
        MySqlCommand command_dr = new MySqlCommand("select
drivers.id, concat(drivers.name, ' ', drivers.surname) as fullname " +
"from drivers left join auto_orders on
drivers.id=auto_orders.iddrivers " +
"where drivers.id not in " +
"(select drivers.id from drivers " +

```

```

        "left join auto_orders on
drivers.id=auto_orders.iddrivers where auto_orders.date = @dt) group by
drivers.id;", db.getConnection()); //auto_orders.date<> @dt
        command_dr.Parameters.Add("@dt", MySqlDbType.String).Value =
dateTimePicker1.Value.ToShortDateString();
        MySqlDataAdapter adapter_drname = new
MySqlDataAdapter(command_dr);
        adapter_drname.Fill(dataSet_dr);
        comboBox2.DataSource = dataSet_dr;
        comboBox2.DisplayMember = "fullname";
        comboBox2.ValueMember = "id";

        DataTable dataSet_tr = new DataTable();
        MySqlCommand command_tr = new MySqlCommand("select auto.id,
concat(auto.mark, ' ;', auto.number, ' ;', auto.kg, ' kr;', auto.volume
, ' m3') as fullname from auto " +
        "left join auto_orders on auto.id=auto_orders.idauto " +
        "where auto.id not in " +
        "(select auto.id from auto " +
        "left join auto_orders on auto.id=auto_orders.idauto
where auto_orders.date = @dt) group by auto.id;", db.getConnection());
//auto_orders.date<> @dt
        command_tr.Parameters.Add("@dt", MySqlDbType.String).Value =
dateTimePicker1.Value.ToShortDateString();
        MySqlDataAdapter adapter_trname = new
MySqlDataAdapter(command_tr);
        adapter_trname.Fill(dataSet_tr);
        comboBox3.DataSource = dataSet_tr;
        comboBox3.DisplayMember = "fullname";
        comboBox3.ValueMember = "id";
        if (comboBox2.Items.Count == 0)
            comboBox2.Enabled = false;
        if (comboBox3.Items.Count == 0)
            comboBox3.Enabled = false;
    }

private void button2_Click(object sender, EventArgs e)
{
    DB db = new DB();
    DataTable table = new DataTable();
    MySqlDataAdapter adapter = new MySqlDataAdapter("select
max(id) from orders;", db.getConnection());
    adapter.Fill(table);
    int orders_id = Convert.ToInt32(table.Rows[0][0]);
    MySqlCommand command2 = new MySqlCommand("insert into
auto_orders (idorders, idauto, iddrivers, date, timestart, timeend)
values (@ido ,@ida, @idd, @dt, @ts, @te)", db.getConnection());
    command2.Parameters.Add("@idd", MySqlDbType.Int32).Value =
Convert.ToInt32(comboBox2.SelectedValue);
    command2.Parameters.Add("@ida", MySqlDbType.Int32).Value =
Convert.ToInt32(comboBox3.SelectedValue);
    command2.Parameters.Add("@ido", MySqlDbType.Int32).Value =
orders_id;
    command2.Parameters.Add("@ts", MySqlDbType.String).Value =
dateTimePicker2.Value.ToString("HH:mm");
    command2.Parameters.Add("@te", MySqlDbType.String).Value =
dateTimePicker3.Value.ToString("HH:mm");
}

```

```

        command2.Parameters.Add("@dt", MySqlDbType.String).Value =
dateTimePicker1.Value.ToShortDateString();
        db.openConnection();
        if (command2.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Договір доповнено!");
        }
        else
            MessageBox.Show("Не вдалося доповнити договір!");
        db.closeConnection();
        combobox_date();
    }
}
}

```

## Лістинг 4 – AddTransportForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class AddTransportForm : Form
    {
        public AddTransportForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
            MySqlCommand command = new MySqlCommand("insert into auto
(mark, number, kg, volume, price) values (@m, @n, @k, @v, @p)",
db.getConnection());
            command.Parameters.Add("@m", MySqlDbType.VarChar).Value =
textBox1.Text;
            command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
textBox2.Text;
            command.Parameters.Add("@k", MySqlDbType.VarChar).Value =
textBox3.Text;
            command.Parameters.Add("@v", MySqlDbType.VarChar).Value =
textBox4.Text;
            command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
textBox5.Text;
            db.openConnection();
            if (command.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Нове авто додано!");
            }
        }
    }
}

```

```

        else
            MessageBox.Show("Не вдалося додати нове авто!");
        db.closeConnection();
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
        textBox5.Clear();
    }
}

```

## Лістинг 5 – DB

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Diplom
{
    class DB
    {
        MySqlConnection connection = new
        MySqlConnection("server=localhost; port=3306; username=root;
        password=k4f128_k4f128; database=autopark");
        public void openConnection()
        {
            if (connection.State == System.Data.ConnectionState.Closed)
                connection.Open();
        }
        public void closeConnection()
        {
            if (connection.State == System.Data.ConnectionState.Open)
                connection.Close();
        }
        public MySqlConnection getConnection()
        {
            return connection;
        }
    }
}

```

## Лістинг 6 – LoginForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```



```

namespace Diplom
{
    public partial class LoginForm : Form
    {
        public LoginForm()
        {
            InitializeComponent();
        }

        private void LoginButton_Click(object sender, EventArgs e)
        {
            Stat.LoginRem = loginField.Text;
            Stat.PassRem = passField.Text;
            string loginUser = loginField.Text;
            string passUser = passField.Text;
            DB db = new DB();
            DataTable table = new DataTable();
            MySqlDataAdapter adapter = new MySqlDataAdapter();
            MySqlCommand command = new MySqlCommand("SELECT * from user
where login = @ul and pass = @up", db.getConnection());
            command.Parameters.Add("@ul", MySqlDbType.VarChar).Value =
loginUser;
            command.Parameters.Add("@up", MySqlDbType.VarChar).Value =
passUser;
            adapter.SelectCommand = command;
            adapter.Fill(table);
            if (table.Rows.Count > 0)
            {
                this.Hide();
                MainForm mainForm = new MainForm();
                mainForm.Show();
                Stat.u_id =
Convert.ToInt32(table.Rows[0]["id"].ToString());
            }
            else
                MessageBox.Show("Невірно введений логін або пароль!");
        }

        private void regOpenButton_Click(object sender, EventArgs e)
        {
            this.Hide();
            RegForm regForm = new RegForm();
            regForm.Show();
        }

        private void LoginForm_FormClosed(object sender,
FormClosedEventArgs e)
        {
            Application.Exit();
        }
    }
}

```

## Лістинг 7 – MainForm

```

using MySql.Data.MySqlClient;
using System;

```

```

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

namespace Diplom
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void MainForm_FormClosed(object sender,
        FormClosedEventArgs e)
        {
            Application.Exit();
        }

        public void MainForm_Load(object sender, EventArgs e)
        {
            DB db = new DB();
            DataSet dataSet_ord = new DataSet();
            DataSet dataSet_cl = new DataSet();
            DataSet dataSet_dr = new DataSet();
            DataSet dataSet_tr = new DataSet();
            MySqlCommand command = new MySqlCommand("select orders.id,
concat(client.name, ' ', client.surname) as namec, " +
            "orders.price, orders.date, " +
            "concat(user.name, ' ', user.surname) as nameu from
orders " +
            "inner join user on orders.userid=user.id " +
            "inner join client on orders.clientid=client.id " +
            "where user.pass=@pa and user.login=@lo",
            db.getConnection());
            command.Parameters.Add("@pa", MySqlDbType.VarChar).Value =
            Stat.PassRem;
            command.Parameters.Add("@lo", MySqlDbType.VarChar).Value =
            Stat.LoginRem;
            MySqlDataAdapter adapter = new MySqlDataAdapter(command);
            adapter.Fill(dataSet_ord);
            ordersView.DataSource = dataSet_ord.Tables[0];
            ordersView.Columns[0].HeaderCell.Value = "ID";
            ordersView.Columns[1].HeaderCell.Value = "ФІО клієнта";
            ordersView.Columns[2].HeaderCell.Value = "Вартість";
            ordersView.Columns[3].HeaderCell.Value = "Дата договору";
            ordersView.Columns[4].HeaderCell.Value = "ФІО диспетчера";

            MySqlDataAdapter adapter_cl = new MySqlDataAdapter("select *
            from client", db.getConnection());
            adapter_cl.Fill(dataSet_cl);
            clientView.DataSource = dataSet_cl.Tables[0];
        }
    }
}

```

```

        MySqlDataAdapter adapter_dr = new MySqlDataAdapter("select *
from drivers", db.getConnection());
        adapter_dr.Fill(dataSet_dr);
        driverView.DataSource = dataSet_dr.Tables[0];

        MySqlDataAdapter adapter_tr = new MySqlDataAdapter("select *
from auto", db.getConnection());
        adapter_tr.Fill(dataSet_tr);
        transportView.DataSource = dataSet_tr.Tables[0];
    }

    private void ExitToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        this.Hide();
        LoginForm loginForm = new LoginForm();
        loginForm.Show();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        AddClientForm addClientForm = new AddClientForm();
        addClientForm.ShowDialog();
        DB db = new DB();
        DataSet dataSet_cl = new DataSet();
        MySqlDataAdapter adapter_cl = new MySqlDataAdapter("select *
from client", db.getConnection());
        adapter_cl.Fill(dataSet_cl);
        clientView.DataSource = null;
        clientView.Refresh();
        clientView.DataSource = dataSet_cl.Tables[0];
    }

    private void button2_Click(object sender, EventArgs e)
    {
        DB db = new DB();
        int selIndex = clientView.SelectedRows[0].Index;
        int rowId = int.Parse(clientView[0,
selIndex].Value.ToString());
        MySqlCommand command = new MySqlCommand("delete from client
where id = @i", db.getConnection());
        command.Parameters.Add("@i", MySqlDbType.VarChar).Value =
rowId;

        clientView.Rows.RemoveAt(selIndex);
        db.openConnection();
        if (command.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Успішно видалено!");
            clientView.Refresh();
        }
        else
            MessageBox.Show("Видалення не виконано!");
        db.closeConnection();
    }

    private void button3_Click(object sender, EventArgs e)
    {

```

```

        int selIndex = clientView.SelectedRows[0].Index;
        Stat.cl_id = int.Parse(clientView[0,
selIndex].Value.ToString());
        Stat.cl_name = clientView[1, selIndex].Value.ToString();
        Stat.cl_surname = clientView[2, selIndex].Value.ToString();
        Stat.cl_phone = clientView[3, selIndex].Value.ToString();
        Stat.cl_mail = clientView[4, selIndex].Value.ToString();
        Stat.cl_osoba = clientView[5, selIndex].Value.ToString();

        RedClientForm redClientForm = new RedClientForm();
        redClientForm.ShowDialog();
        DB db = new DB();
        DataSet dataSet_cl = new DataSet();
        MySqlDataAdapter adapter_cl = new MySqlDataAdapter("select *
from client", db.getConnection());
        adapter_cl.Fill(dataSet_cl);
        clientView.DataSource = null;
        clientView.Refresh();
        clientView.DataSource = dataSet_cl.Tables[0];
    }

    private void button6_Click(object sender, EventArgs e)
    {
        AddDriverForm addDriverForm = new AddDriverForm();
        addDriverForm.ShowDialog();
        DB db = new DB();
        DataSet dataSet_dr = new DataSet();
        MySqlDataAdapter adapter_dr = new MySqlDataAdapter("select *
from drivers", db.getConnection());
        adapter_dr.Fill(dataSet_dr);
        driverView.DataSource = null;
        driverView.Refresh();
        driverView.DataSource = dataSet_dr.Tables[0];
    }

    private void button5_Click(object sender, EventArgs e)
    {
        DB db = new DB();
        int selIndex = driverView.SelectedRows[0].Index;
        int rowId = int.Parse(driverView[0,
selIndex].Value.ToString());
        MySqlCommand command = new MySqlCommand("delete from drivers
where id = @i", db.getConnection());
        command.Parameters.Add("@i", MySqlDbType.VarChar).Value =
rowId;

        driverView.Rows.RemoveAt(selIndex);
        db.openConnection();
        if (command.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Успішно видалено!");
            driverView.Refresh();
        }
        else
            MessageBox.Show("Видалення не виконано!");
        db.closeConnection();
    }
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    int selIndex = driverView.SelectedRows[0].Index;
    Stat.dr_id = int.Parse(driverView[0,
selIndex].Value.ToString());
    Stat.dr_name = driverView[1, selIndex].Value.ToString();
    Stat.dr_surname = driverView[2, selIndex].Value.ToString();
    Stat.dr_phone = driverView[3, selIndex].Value.ToString();
    Stat.dr_mail = driverView[4, selIndex].Value.ToString();

    RedDriverForm redDriverForm = new RedDriverForm();
    redDriverForm.ShowDialog();
    DB db = new DB();
    DataSet dataSet_dr = new DataSet();
    MySqlDataAdapter adapter_dr = new MySqlDataAdapter("select *
from drivers", db.getConnection());
    adapter_dr.Fill(dataSet_dr);
    driverView.DataSource = null;
    driverView.Refresh();
    driverView.DataSource = dataSet_dr.Tables[0];
}

private void button9_Click(object sender, EventArgs e)
{
    AddTransportForm addTransportForm = new AddTransportForm();
    addTransportForm.ShowDialog();
    DB db = new DB();
    DataSet dataSet_tr = new DataSet();
    MySqlDataAdapter adapter_tr = new MySqlDataAdapter("select *
from auto", db.getConnection());
    adapter_tr.Fill(dataSet_tr);
    transportView.DataSource = null;
    transportView.Refresh();
    transportView.DataSource = dataSet_tr.Tables[0];
}

private void button8_Click(object sender, EventArgs e)
{
    DB db = new DB();
    int selIndex = transportView.SelectedRows[0].Index;
    int rowId = int.Parse(transportView[0,
selIndex].Value.ToString());
    MySqlCommand command = new MySqlCommand("delete from auto
where id = @i", db.getConnection());
    command.Parameters.Add("@i", MySqlDbType.VarChar).Value =
rowId;

    transportView.Rows.RemoveAt(selIndex);
    db.openConnection();
    if (command.ExecuteNonQuery() == 1)
    {
        MessageBox.Show("Успішно видалено!");
        transportView.Refresh();
    }
    else
        MessageBox.Show("Видалення не виконано!");
    db.closeConnection();
}

```

```

private void button7_Click(object sender, EventArgs e)
{
    int selIndex = transportView.SelectedRows[0].Index;
    Stat.tr_id = int.Parse(transportView[0,
selIndex].Value.ToString());
    Stat.tr_mark = transportView[1, selIndex].Value.ToString();
    Stat.tr_number = transportView[2,
selIndex].Value.ToString();
    Stat.tr_kg = int.Parse(transportView[3,
selIndex].Value.ToString());
    Stat.tr_volume = transportView[4,
selIndex].Value.ToString();
    Stat.tr_price = transportView[5, selIndex].Value.ToString();

    RedTransportForm redTransportForm = new RedTransportForm();
    redTransportForm.ShowDialog();
    DB db = new DB();
    DataSet dataSet_tr = new DataSet();
    MySqlDataAdapter adapter_tr = new MySqlDataAdapter("select *
from auto", db.getConnection());
    adapter_tr.Fill(dataSet_tr);
    transportView.DataSource = null;
    transportView.Refresh();
    transportView.DataSource = dataSet_tr.Tables[0];
}

private void button12_Click(object sender, EventArgs e)
{
    AddOrdersForm addOrdersForm = new AddOrdersForm();
    DB db = new DB();
    DataTable dataSet_cl = new DataTable();
    MySqlDataAdapter adapter_clname = new
MySqlDataAdapter("select id, concat(name, ' ', surname) as fullname from
client;", db.getConnection());
    adapter_clname.Fill(dataSet_cl);
    addOrdersForm.comboBox1.DataSource = dataSet_cl;
    addOrdersForm.comboBox1.DisplayMember = "fullname";
    addOrdersForm.comboBox1.ValueMember = "id";
    //Stat.vm_c =
Convert.ToInt32(addOrdersForm.comboBox1.SelectedValue);

    /*DataTable dataSet_dr = new DataTable();
    MySqlDataAdapter adapter_drname = new
MySqlDataAdapter("select id, concat(name, ' ', surname) as fullname from
drivers;", db.getConnection());
    adapter_drname.Fill(dataSet_dr);
    addOrdersForm.comboBox2.DataSource = dataSet_dr;
    addOrdersForm.comboBox2.DisplayMember = "fullname";
    addOrdersForm.comboBox2.ValueMember = "id";*/
    //Stat.vm_d =
Convert.ToInt32(addOrdersForm.comboBox2.SelectedValue);

    /*DataTable dataSet_tr = new DataTable();
    MySqlDataAdapter adapter_trname = new
MySqlDataAdapter("select id, concat(mark, ' ;', number, ' ;', kg, '
kr;', volume, ' m3') as fullname from auto;", db.getConnection());
    adapter_trname.Fill(dataSet_tr);
    addOrdersForm.comboBox3.DataSource = dataSet_tr;

```

```

        addOrdersForm.comboBox3.DisplayMember = "fullname";
        addOrdersForm.comboBox3.ValueMember = "id";*/
        //Stat.vm_a =
Convert.ToInt32(addOrdersForm.comboBox3.SelectedValue);
        addOrdersForm.ShowDialog();
        DataSet dataSet_ord = new DataSet();
        MySqlCommand command = new MySqlCommand("select orders.id,
concat(client.name, ' ', client.surname) as namec, " +
        "orders.price, orders.date, " +
        "concat(user.name, ' ', user.surname) as nameu from
orders " +
        "inner join user on orders.userid=user.id " +
        "inner join client on orders.clientid=client.id " +
        "where user.pass=@pa and user.login=@lo",
db.getConnection());
        command.Parameters.Add("@pa", MySqlDbType.VarChar).Value =
Stat.PassRem;
        command.Parameters.Add("@lo", MySqlDbType.VarChar).Value =
Stat.LoginRem;
        MySqlDataAdapter adapter = new MySqlDataAdapter(command);
        adapter.Fill(dataSet_ord);
        ordersView.DataSource = dataSet_ord.Tables[0];
    }

    private void button11_Click(object sender, EventArgs e)
    {
        DB db = new DB();
        int selIndex = ordersView.SelectedRows[0].Index;
        int rowId = int.Parse(ordersView[0,
selIndex].Value.ToString());
        MySqlCommand command = new MySqlCommand("delete from orders
where id = @i", db.getConnection());
        command.Parameters.Add("@i", MySqlDbType.VarChar).Value =
rowId;

        ordersView.Rows.RemoveAt(selIndex);
        db.openConnection();
        if (command.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Успішно видалено!");
            ordersView.Refresh();
        }
        else
            MessageBox.Show("Видалення не виконано!");
        db.closeConnection();
    }

    private void ordersView_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
    {
        int selIndex = ordersView.SelectedRows[0].Index;
        Stat.or_id = int.Parse(ordersView[0,
selIndex].Value.ToString());

        InfoOrderForm infoOrderForm = new InfoOrderForm();
        infoOrderForm.ShowDialog();
    }

```

```

    }
}

```

## Лістинг 8 – Program

```

using System;

using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new LoginForm());
        }
    }
}

```

## Лістинг 9 – RedClientForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class RedClientForm : Form
    {
        public RedClientForm()
        {
            InitializeComponent();
        }

        private void RedClientForm_Load(object sender, EventArgs e)
        {
            textBox1.Text = Stat.cl_name;
            textBox2.Text = Stat.cl_surname;
            textBox3.Text = Stat.cl_phone;
            textBox4.Text = Stat.cl_mail;
            comboBox1.Text = Stat.cl_osoba;
        }
    }
}

```



```

        if (comboBox1.Text=="Юридична")
        {
            label2.Visible = false;
            textBox2.Visible = false;
            label11.Visible = false;
            label6.Visible = true;
            textBox2.Text = "-----";
        }
        else
        {
            label2.Visible = true;
            textBox2.Visible = true;
            label6.Visible = false;
            label11.Visible = true;
            textBox2.Clear();
        }
    }

    private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (comboBox1.SelectedIndex == 1)
        {
            label2.Visible = false;
            textBox2.Visible = false;
            label11.Visible = false;
            label6.Visible = true;
            textBox2.Text = "-----";
        }
        else
        {
            label2.Visible = true;
            textBox2.Visible = true;
            label6.Visible = false;
            label11.Visible = true;
            textBox2.Clear();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        DB db = new DB();
        MySqlCommand command = new MySqlCommand("update client set
name = @n, surname = @s, phone=@p, mail=@m, osoba=@o where id=@id",
db.getConnection());
        command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
textBox1.Text;
        command.Parameters.Add("@s", MySqlDbType.VarChar).Value =
textBox2.Text;
        command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
textBox3.Text;
        command.Parameters.Add("@m", MySqlDbType.VarChar).Value =
textBox4.Text;
        command.Parameters.Add("@o", MySqlDbType.VarChar).Value =
comboBox1.Text;
        command.Parameters.Add("@id", MySqlDbType.Int32).Value =
Stat.cl_id;
        db.openConnection();
    }

```

```

        if (command.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Редагування успішне!");
        }
        else
            MessageBox.Show("Не вдалося додати нового користувача!");
        db.closeConnection();
        this.Hide();
    }
}
}

```

## Лістинг 10 – RedDriverForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class RedDriverForm : Form
    {
        public RedDriverForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
            MySqlCommand command = new MySqlCommand("update drivers set
name = @n, surname = @s, phone=@p, mail=@m where id=@id",
db.getConnection());
            command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
textBox1.Text;
            command.Parameters.Add("@s", MySqlDbType.VarChar).Value =
textBox2.Text;
            command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
textBox3.Text;
            command.Parameters.Add("@m", MySqlDbType.VarChar).Value =
textBox4.Text;
            command.Parameters.Add("@id", MySqlDbType.Int32).Value =
Stat.dr_id;
            db.openConnection();
            if (command.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Редагування успішне!");
            }
            else

```

```

        MessageBox.Show("Не вдалося додати нового
користувача!");
        db.closeConnection();
        this.Hide();
    }

    private void RedDriverForm_Load(object sender, EventArgs e)
    {
        textBox1.Text = Stat.dr_name;
        textBox2.Text = Stat.dr_surname;
        textBox3.Text = Stat.dr_phone;
        textBox4.Text = Stat.dr_mail;
    }
}

```

## Лістинг 11 – RedOrdersForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class AddOrdersForm : Form
    {
        public AddOrdersForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
            string date= DateTime.Now.ToShortDateString();
            MySqlCommand command = new MySqlCommand("insert into orders
(userid, clientid, date) values (@ui, @ci, @date)", db.getConnection());
            command.Parameters.Add("@ci", MySqlDbType.Int32).Value =
Convert.ToInt32(comboBox1.SelectedValue);
            command.Parameters.Add("@ui", MySqlDbType.Int32).Value =
Stat.u_id;
            command.Parameters.Add("@date", MySqlDbType.String).Value =
date;
            db.openConnection();
            if (command.ExecuteNonQuery() == 1)
            {
                DataTable table = new DataTable();
                MySqlDataAdapter adapter = new MySqlDataAdapter("select
max(id) from orders;", db.getConnection());
                adapter.Fill(table);
                int orders_id = Convert.ToInt32(table.Rows[0][0]);
            }
        }
    }
}

```

```

        MySqlCommand command2 = new MySqlCommand("insert into
auto_orders (idorders, idauto, iddrivers, date, timestart, timeend)
values (@ido, @ida, @idd, @dt, @ts, @te)", db.getConnection());
        command2.Parameters.Add("@idd", MySqlDbType.Int32).Value
= Convert.ToInt32(comboBox2.SelectedValue);
        command2.Parameters.Add("@ida", MySqlDbType.Int32).Value
= Convert.ToInt32(comboBox3.SelectedValue);
        command2.Parameters.Add("@ido", MySqlDbType.Int32).Value
= orders_id;
        command2.Parameters.Add("@ts", MySqlDbType.String).Value
= dateTimePicker2.Value.ToString("HH:mm");
        command2.Parameters.Add("@te", MySqlDbType.String).Value
= dateTimePicker3.Value.ToString("HH:mm");
        command2.Parameters.Add("@dt", MySqlDbType.String).Value
= dateTimePicker1.Value.ToShortDateString();
        if (command2.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Новий контракт створено!");
        }
        else
            MessageBox.Show("Не вдалося додати новий запис до
контракту!");
    }
    else
        MessageBox.Show("Не вдалося створити новий контракт!");
    db.closeConnection();
    combobox_date();
}

private void dateTimePicker1_ValueChanged(object sender,
EventArgs e)
{
    combobox_date();
}

private void combobox_date()
{
    comboBox2.Enabled = true;
    comboBox3.Enabled = true;
    DB db = new DB();
    DataTable dataSet_dr = new DataTable();
    MySqlCommand command_dr = new MySqlCommand("select
drivers.id, concat(drivers.name, ' ', drivers.surname) as fullname " +
"from drivers left join auto_orders on
drivers.id=auto_orders.iddrivers " +
"where drivers.id not in " +
"(select drivers.id from drivers " +
"left join auto_orders on
drivers.id=auto_orders.iddrivers where auto_orders.date = @dt) group by
drivers.id;", db.getConnection()); //auto_orders.date<> @dt
    command_dr.Parameters.Add("@dt", MySqlDbType.String).Value =
dateTimePicker1.Value.ToShortDateString();
    MySqlDataAdapter adapter_drname = new
MySqlDataAdapter(command_dr);
    adapter_drname.Fill(dataSet_dr);
    comboBox2.DataSource = dataSet_dr;

```

```

        comboBox2.DisplayMember = "fullname";
        comboBox2.ValueMember = "id";

        DataTable dataSet_tr = new DataTable();
        MySqlCommand command_tr = new MySqlCommand("select auto.id,
concat(auto.mark, ' ;', auto.number, ' ;', auto.kg, ' кг;', auto.volume
, ' m3') as fullname from auto " +
        "left join auto_orders on auto.id=auto_orders.idauto " +
        "where auto.id not in " +
        "(select auto.id from auto " +
        "left join auto_orders on auto.id=auto_orders.idauto
where auto_orders.date = @dt) group by auto.id;", db.getConnection());
//auto_orders.date<> @dt
        command_tr.Parameters.Add("@dt", MySqlDbType.String).Value =
dateTimePicker1.Value.ToShortDateString();
        MySqlDataAdapter adapter_trname = new
MySqlDataAdapter(command_tr);
        adapter_trname.Fill(dataSet_tr);
        comboBox3.DataSource = dataSet_tr;
        comboBox3.DisplayMember = "fullname";
        comboBox3.ValueMember = "id";
        if (comboBox2.Items.Count == 0)
            comboBox2.Enabled = false;
        if (comboBox3.Items.Count == 0)
            comboBox3.Enabled = false;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        DB db = new DB();
        DataTable table = new DataTable();
        MySqlDataAdapter adapter = new MySqlDataAdapter("select
max(id) from orders;", db.getConnection());
        adapter.Fill(table);
        int orders_id = Convert.ToInt32(table.Rows[0][0]);
        MySqlCommand command2 = new MySqlCommand("insert into
auto_orders (idorders, idauto, iddrivers, date, timestart, timeend)
values (@ido ,@ida, @idd, @dt, @ts, @te)", db.getConnection());
        command2.Parameters.Add("@idd", MySqlDbType.Int32).Value =
Convert.ToInt32(comboBox2.SelectedValue);
        command2.Parameters.Add("@ida", MySqlDbType.Int32).Value =
Convert.ToInt32(comboBox3.SelectedValue);
        command2.Parameters.Add("@ido", MySqlDbType.Int32).Value =
orders_id;
        command2.Parameters.Add("@ts", MySqlDbType.String).Value =
dateTimePicker2.Value.ToString("HH:mm");
        command2.Parameters.Add("@te", MySqlDbType.String).Value =
dateTimePicker3.Value.ToString("HH:mm");
        command2.Parameters.Add("@dt", MySqlDbType.String).Value =
dateTimePicker1.Value.ToShortDateString();
        db.openConnection();
        if (command2.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Договір доповнено!");
        }
        else
            MessageBox.Show("Не вдалося доповнити договір!");
        db.closeConnection();
    }

```

```

        combobox_date();
    }
}

```

## Лістинг 12 – RedTransportForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class RedTransportForm : Form
    {
        public RedTransportForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DB db = new DB();
            MySqlCommand command = new MySqlCommand("update auto set
mark = @m, number = @n, kg=@k, volume=@v, price=@p where id=@id",
db.getConnection());
            command.Parameters.Add("@m", MySqlDbType.VarChar).Value =
textBox1.Text;
            command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
textBox2.Text;
            command.Parameters.Add("@k", MySqlDbType.VarChar).Value =
textBox3.Text;
            command.Parameters.Add("@v", MySqlDbType.VarChar).Value =
textBox4.Text;
            command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
textBox5.Text;
            command.Parameters.Add("@id", MySqlDbType.Int32).Value =
Stat.tr_id;
            db.openConnection();
            if (command.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Редагування успішне!");
            }
            else
                MessageBox.Show("Не вдалося додати нового
користувача!");
            db.closeConnection();
            this.Hide();
        }

        private void RedTransportForm_Load(object sender, EventArgs e)

```

```

        {
            textBox1.Text = Stat.tr_mark;
            textBox2.Text = Stat.tr_number;
            textBox3.Text = Stat.tr_kg.ToString();
            textBox4.Text = Stat.tr_volume;
            textBox5.Text = Stat.tr_price;
        }
    }
}

```

## Лістинг 12 – RegForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class RegForm : Form
    {
        public RegForm()
        {
            InitializeComponent();
        }

        private void RegForm_Load(object sender, EventArgs e)
        {
            nameField.Text = "Ім'я...";
            nameField.ForeColor = Color.Gray;
            surnameField.Text = "Прізвище...";
            surnameField.ForeColor = Color.Gray;
            phoneField.Text = "Телефон...";
            phoneField.ForeColor = Color.Gray;
            loginField.Text = "Логін...";
            loginField.ForeColor = Color.Gray;
            passField.Text = "Пароль...";
            passField.ForeColor = Color.Gray;
            passField.UseSystemPasswordChar = false;
            label1.Select();
        }

        private void nameField_Enter(object sender, EventArgs e)
        {
            if(nameField.Text=="Ім'я...")
            {
                nameField.Text = null;
                nameField.ForeColor = Color.Black;
            }
        }

        private void nameField_Leave(object sender, EventArgs e)

```

```

{
    if (nameField.Text == "")
    {
        nameField.Text = "Ім'я...";
        nameField.ForeColor = Color.Gray;
    }
}

private void surnameField_Enter(object sender, EventArgs e)
{
    if (surnameField.Text == "Прізвище...")
    {
        surnameField.Text = null;
        surnameField.ForeColor = Color.Black;
    }
}

private void surnameField_Leave(object sender, EventArgs e)
{
    if (surnameField.Text == "")
    {
        surnameField.Text = "Прізвище...";
        surnameField.ForeColor = Color.Gray;
    }
}

private void phoneField_Enter(object sender, EventArgs e)
{
    if (phoneField.Text == "Телефон...")
    {
        phoneField.Text = null;
        phoneField.ForeColor = Color.Black;
    }
}

private void phoneField_Leave(object sender, EventArgs e)
{
    if (phoneField.Text == "")
    {
        phoneField.Text = "Телефон...";
        phoneField.ForeColor = Color.Gray;
    }
}

private void loginField_Enter(object sender, EventArgs e)
{
    if (loginField.Text == "Логін...")
    {
        loginField.Text = null;
        loginField.ForeColor = Color.Black;
    }
}

private void loginField_Leave(object sender, EventArgs e)
{
    if (loginField.Text == "")
    {
        loginField.Text = "Логін...";
    }
}

```



```

        loginField.ForeColor = Color.Gray;
    }
}

private void passField_Enter(object sender, EventArgs e)
{
    if (passField.Text == "Пароль...")
    {
        passField.Text = null;
        passField.ForeColor = Color.Black;
        passField.UseSystemPasswordChar = true;
    }
}

private void passField_Leave(object sender, EventArgs e)
{
    if (passField.Text == "")
    {
        passField.Text = "Пароль...";
        passField.ForeColor = Color.Gray;
        passField.UseSystemPasswordChar = false;
    }
}

public Boolean IsUserExists()
{
    DB db = new DB();
    DataTable table = new DataTable();
    MySqlDataAdapter adapter = new MySqlDataAdapter();
    MySqlCommand command = new MySqlCommand("select * from user
where login = @1", db.getConnection());
    command.Parameters.Add("@1", MySqlDbType.VarChar).Value =
loginField.Text;
    adapter.SelectCommand = command;
    adapter.Fill(table);
    if(table.Rows.Count>0)
    {
        MessageBox.Show("Такий логін вже існує!");
        return true;
    }
    else
        return false;
}

private void regButton_Click(object sender, EventArgs e)
{
    if(nameField.Text=="Ім'я...")
    {
        MessageBox.Show("Введіть ім'я!");
        return;
    }
    if(surnameField.Text == "Прізвище...")
    {
        MessageBox.Show("Введіть прізвище!");
        return;
    }
    if (phoneField.Text == "Телефон...")
    {
        MessageBox.Show("Введіть номер телефону!");
    }
}

```

```

        return;
    }
    if (loginField.Text == "Логін...")
    {
        MessageBox.Show("Введіть логін!");
        return;
    }
    if (passField.Text == "Пароль...")
    {
        MessageBox.Show("Введіть пароль!");
        return;
    }
    if (IsUserExists())
        return;
    DB db = new DB();
    MySqlCommand command = new MySqlCommand("insert into user
(name, surname, phone, login, pass) values (@n, @s, @p, @l, @pas)",
db.getConnection());
    command.Parameters.Add("@n", MySqlDbType.VarChar).Value =
nameField.Text;
    command.Parameters.Add("@s", MySqlDbType.VarChar).Value =
surnameField.Text;
    command.Parameters.Add("@p", MySqlDbType.VarChar).Value =
phoneField.Text;
    command.Parameters.Add("@l", MySqlDbType.VarChar).Value =
loginField.Text;
    command.Parameters.Add("@pas", MySqlDbType.VarChar).Value =
passField.Text;
    db.openConnection();
    if (command.ExecuteNonQuery() == 1)
    {
        MessageBox.Show("Реєстрація успішна!");
        this.Hide();
        LoginForm loginForm = new LoginForm();
        loginForm.Show();
    }
    else
        MessageBox.Show("Реєстрація не виконана!");
    db.closeConnection();
}

private void backButton_Click(object sender, EventArgs e)
{
    LoginForm loginForm = new LoginForm();
    loginForm.Show();
    this.Hide();
}

private void RegForm_FormClosed(object sender,
FormClosedEventArgs e)
{
    Application.Exit();
}
}
}

```

### Лістинг 13 – InfoOrderForm

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diplom
{
    public partial class InfoOrderForm : Form
    {
        public InfoOrderForm()
        {
            InitializeComponent();
        }

        private void InfoOrderForm_Load(object sender, EventArgs e)
        {
            DB db = new DB();
            DataSet dataSet_inf = new DataSet();
            MySqlCommand command = new MySqlCommand("select
auto_orders.idorders, concat(drivers.name, ' ', drivers.surname) as
named, auto.mark, auto.number, concat(auto_orders.date, ' ',
auto_orders.timestart, ' - ', auto_orders.timeend) as datetime " +
            "from auto_orders left join drivers on
auto_orders.iddrivers=drivers.id " +
            "left join auto on auto_orders.idauto=auto.id where
auto_orders.idorders=@id", db.getConnection());
            command.Parameters.Add("@id", MySqlDbType.Int32).Value =
Stat.or_id;
            MySqlDataAdapter adapter_inf = new
MySqlDataAdapter(command);
            adapter_inf.Fill(dataSet_inf);
            ordersView.DataSource = dataSet_inf.Tables[0];
            ordersView.Columns[0].HeaderCell.Value = "ID";
            ordersView.Columns[1].HeaderCell.Value = "ФІО водія";
            ordersView.Columns[2].HeaderCell.Value = "Модель";
            ordersView.Columns[3].HeaderCell.Value = "Державний номер";
            ordersView.Columns[4].HeaderCell.Value = "Дата та час
роботи";
        }
    }
}

```

**ВІДГУК**

**керівника економічного розділу  
на кваліфікаційну роботу бакалавра  
на тему:**

**«Розробка інформаційної автоматизованої системи для роботи диспетчера  
сервісу оренди автотранспорту для вантажоперевезень»  
студента групи 122-19ск-2 Часника Владислава Сергійовича**

**Керівник економічного розділу  
доцент каф. ПЕП та ПУ, к.е.н**

**Л. В. Касьяненко**

## ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файла	Опис
Пояснювальні документи	
Диплом.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом.pdf	Пояснювальна записка до кваліфікаційної роботи. в форматі PDF
Програма	
Diplom.zip	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація кваліфікаційної роботи.