Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “Об’єктно-орієнтоване програмування”

за темою “Автосалон”

Зображення, що містить ескіз, малюнок, картинки, Лінійне мистецтво

Вміст, створений ШІ, може бути неправильним.Керівник , проф. Бондарєв В.М.

Студент гр. ПЗПІ-24-1 Бабіч О. І.

Комісія:

Проф. Бондарєв В.М.

Ст. викл. Черепанова Ю.Ю.

Ст. викл. Ляпота В.М.

Харків 2025

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра *програмної інженерії*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об’єктно-орієнтоване програмування*

Спеціальність *121 Інженерія програмного забезпечення*

Освітня програма: *Програмна інженерія*

Курс *1* . Група *ПЗПІ-24-1*. Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проєкт студента***

*Бабіч Олександра Ігоровича*

1 Тема проєкту:

*Автосалон*

2 Термін здачі студентом закінченого проєкту: ***“07” червня 2025 р.***

3 Вихідні дані проєкту:

*База нових і старих автомобілів: марка, модель, рік випуску, технічні характеристики, запитувана ціна. База покупців: контактні дані, вимоги до марки, моделі, та технічного стану, фінансові можливості. Автоматизація підбору варіантів для покупця, формування заявки для постачальників.*

Зміст розрахунково-пояснювальної записки:

*Вступ, опис вимог, проєктування програми, інструкція користувача, висновки*

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження  і затвердження теми (2 тижд.) | 03 - 03 - 2025 р. – 14 - 03 - 2025 р. |
| 2 | Формулювання мети роботи (1 тижд.) | 17-03-2025 – 21-03-2025 р. |
| 3 | Опис вимог до програми (1-2 тижд.) | 24-03-2025 – 04-04-2025 р |
| 4 | Проєктування програми (1-2 тижд.) | 07-04-2025 – 18-04-2025 р |
| 5 | Кодування (3 тижд.) | 21-04-2025 – 09-05-2025 р |
| 6 | Створення інструкції користувача (1 тижд.) | 12-05-2025 – 16-05-2025 р |
| 7 | Підготовка роботи до захисту (1 тижд.) | 26-05-2025 – 02-06-2025 р |
| 8 | Захист | 19-05-2025 – 07- 06 -2025 р |

Зображення, що містить ескіз, малюнок, картинки, Лінійне мистецтво

Вміст, створений ШІ, може бути неправильним.

Студент гр. ПЗПІ-24-1 Бабіч О. І.

Керівник, проф. Бондарєв В. М.

« 3 » березня\_\_\_\_\_ 2025 р

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 44 с., 10 рис., 3 табл., 2 додатки, 7 джерел.

ЗВІТ, АВТОСАЛОН, МОВА C#, ООП, ПОКУПЕЦЬ,.NET, WINDOWS FORMS

Метою роботи є розробка програми “Auto Matcher”, яка призначена для процесу пошуку та придбання автомобілів в автосалоні.

У результаті розробки створено програмне забезпечення “Auto Matcher”, призначене для автоматизації процесу пошуку та придбання транспортних засобів в автосалоні “Top Diller”. Характеристиками кожного авто є: марка, модель, країна виробництва, рік випуску, тип двигуна, потужність, тип пального та вартість. Програма забезпечує користувачам доступ до актуальної інформації про всі автомобілі, наявні в автосалоні, а також дозволяють формувати заявки на придбання обраних моделей.

В процесі розробки використано середовище розробки Microsoft Visual Studio 2022, фреймворк Windows Forms, платформа .NET 8.0, мова програмування C#.

**ЗМІСТ**

***//Ще не сформовано***

Вступ ....................................................................................................... 7

1 Опис вимог............................................................................................ 8

1.1 Сценарії використання .......................................................... 8

1.2 Функції програми...................................................................10

2 Проєктування програми. .....................................................................12

2.1 Назва першого підрозділу ......................................................12

2.2 Назва другого підрозділу .......................................................16

2.3 Назва третього підрозділу.......................................................18

3 Інструкція користувача.........................................................................22

Висновки....................................................................................................26

Перелік джерел посилання.....................................................................27

Додаток А Назва додатка.. ................................................................... 28

# ВСТУП

Метою розробки програмного забезпечення “Auto Matcher” є створення автоматизованої системи керування діяльністю автосалону. Програмне забезпечення підтримує роботу з базами даних автомобілів, покупців та заявок на викуп авто.

# ОПИС ВИМОГ

* 1. Сценарії використання
     1. Сценарій “Пошук автомобіля”

Передумова

Користувач відкрив головне вікно програми.

Основний сценарій:

1. Користувач заповнює поля на панелі пошуку і натискає кнопку “Find”.
2. Програма знаходить автомобілі, які відповідають умовам пошуку.
3. Список знайдених автомобілів з’являється на екрані.

Додатковий сценарій:

1. Користувач натискає кнопку “Find”, не заповнюючи поля.
2. Програма відображає список усіх автомобілів.
3. Список усіх автомобілів з’являється на екрані.

Додатковий сценарій:

1. Користувач заповнює поля на панелі пошуку і натискає кнопку “Find”.
2. Програма не знаходить жодного автомобіля, який відповідає умовам пошуку.
3. Користувач бачить повідомлення, що автомобілів не знайдено.
   * 1. Сценарій “Реєстрація нового користувача”

Передумова

Користувач відкрив головне вікно програми та натиснув кнопку “Sign up”.

Основний сценарій:

1. Програма відкриває вікно реєстрації нового користувача.
2. Користувач заповнює усі текстові поля.
3. Користувач натискає на кнопку “Sign up”.
4. Програма перевіряє на унікальність введений користувачем логін, обходячи список зареєстрованих користувачів.
5. Програма перевіряє коректність введених даних.
6. Програма сповіщає користувача про успіх операції реєстрації.

Додатковий сценарій:

1. Програма відкриває вікно реєстрації нового користувача.
2. Користувач заповнює усі текстові поля.
3. Користувач натискає на кнопку “Sign up”.
4. Програма встановлює, що у списку зареєстрованих користувачів вже існує користувач з логіном, введеним в текстове поле для логіну.
5. Програма сповіщає користувача про наявність облікового запису з таким самим логіном.

Додатковий сценарій:

1. Програма відкриває вікно реєстрації нового користувача.
2. Користувач не заповнює жодного з текстових полів.
3. Користувач натискає на кнопку “Sign up”.
4. Програма виявляє, що введені дані є некоректними.
5. Програма сповіщає користувача про помилку в даних та надає інструкції щодо виправлення.
   * 1. Сценарій “Авторизація користувача”

Передумова

Користувач відкрив головне вікно програми та натиснув кнопку “Log in”.

Основний сценарій:

1. Програма відкриває вікно авторизації.
2. Користувач заповнює текстові поля логіну та паролю.
3. Користувач натискає на кнопку “Log in”.
4. Програма перевіряє список користувачів та знаходить обліковий запис, що відповідає введеним даним.
5. Програма сповіщає користувача про успіх операції авторизації та повертає на головне вікно програми.

Додатковий сценарій:

1. Програма відкриває вікно авторизації.
2. Користувач заповнює текстові поля логіну та паролю.
3. Користувач натискає на кнопку “Log in”.
4. Програма не знаходить облікового запису з відповідними даними.
5. Програма сповіщає користувача про помилку авторизації.
   * 1. Сценарій “Автоматизований пошук автомобіля”

Передумова

Користувач успішно пройшов операцію авторизації.

Основний сценарій:

1. Користувач натискає кнопку “Auto Find”.
2. Програма аналізує умови, які були вказані користувачем під час реєстрації.
3. Програма повертає список автомобілів, які відповідають цим умовам.
4. Список знайдених автомобілів з’являється на екрані.

Додатковий сценарій:

1. Користувач натискає кнопку “Auto Find”.
2. Програма визначає, що під час реєстрації користувач не вказав необхідні дані для автоматизованого пошуку.
3. Програма сповіщає користувача про відсутність автомобілів, що відповідають заданим умовам.

Додатковий сценарій:

1. Користувач натискає кнопку “Auto Find”.
2. Програма не знаходить жодного автомобіля, який задовольняє умовам, введеним користувачем при реєстрації.
3. Програма сповіщає користувача про відсутність автомобілів, що відповідають заданим умовам.
   * 1. Сценарій “Формування заяви на викуп автомобілів”

Передумова

Користувач успішно пройшов операцію авторизації.

Основний сценарій:

1. Користувач обирає автомобілі, що його цікавлять та натискає на кнопку “New Application”.
2. Користувач бачить повідомлення про успішно сформовано заяву та ID заяви.

Додатковий сценарій:

1. Користувач не обирає жодного авто та натискає кнопку “New Application”.
2. Користувач бачить повідомлення про необхідність вибору автомобілів для формування заяви на викуп.
   * 1. Сценарій “Перегляд надісланих заяв користувачем”

Передумова

Користувач успішно пройшов операцію авторизації.

Основний сценарій:

1. Користувач натискає на кнопку “My Applications”.
2. Програма аналізує список заяв, та повертає лише ті, в яких використовується логін користувача, який зараз використовує програму.
3. Програма відображає список заяв у спеціально відведеній області.

Додатковий сценарій:

1. Користувач натискає на кнопку “My Applications”.
2. Програма не знаходить жодної заяви, надісланої користувачем.
3. Програма відображає повідомлення про відсутність надісланих заяв у спеціально відведеній області.
   * 1. Сценарій “Відкриття панелі адміністратора”

Передумова

Адміністратор відкрив головне вікно програми.

Основний сценарій:

1. Адміністратор натискає на кнопку “Log in”.
2. Адміністратор вводить спеціальну комбінацію логіна та пароля.
3. Програма перевіряє відповідність введених даних.
4. У разі успіху програма відкриває панель адміністратора програми.
   * 1. Сценарій “Додавання нового автомобіля до списку”

Передумова

Адміністратор відкрив панель.

Основний сценарій:

1. Адміністратор відкриває вкладку з автомобілями.
2. Адміністратор натискає на кнопку “Add ”.
3. Програма відкриває вікно додавання автомобіля.
4. Адміністратор заповнює текстові поля та натискає на кнопку “Add Car”.
5. Програма перевіряє на коректність введені дані.
6. Програма додає до списку автомобілів новий.

Додатковий сценарій:

1. Адміністратор відкриває вкладку з автомобілями.
2. Адміністратор натискає на кнопку “Add ”.
3. Програма відкриває вікно додавання автомобіля.
4. Адміністратор вводить некоректні дані в текстові поля.
5. Програма встановлює, що введені дані є некоректними.
6. Програма сповіщає адміністратора про помилку у введених даних та спосіб її виправлення.
   * 1. Сценарій “Редагування даних автомобіля”

Передумова

Адміністратор відкрив панель.

Основний сценарій:

1. Адміністратор відкриває вкладку з автомобілями.
2. Адміністратор обирає автомобіль зі списку.
3. Адміністратор натискає кнопку “Edit”.
4. Програма відкриває вікно редагування автомобіля з даними обраного авто.
5. Адміністратор редагує дані автомобіля.
6. Програма зберігає оновлені дані автомобіля.
   * 1. Сценарій “Видалення автомобіля зі списку”

Передумова

Адміністратор відкрив панель.

Основний сценарій:

1. Адміністратор відкриває вкладку з автомобілями.
2. Адміністратор обирає автомобіль зі списку.
3. Адміністратор натискає кнопку “Delete”.
4. Програма просить підтвердження дії.
5. Адміністратор підтверджує дію.
6. Програма видаляє обраний автомобіль зі списку.

Додатковий сценарій

1. Адміністратор відкриває вкладку з автомобілями.
2. Адміністратор обирає автомобіль зі списку.
3. Адміністратор натискає кнопку “Delete”.
4. Програма просить підтвердження дії.
5. Адміністратор cкасовує дію.
6. Програма скасовує операцію видалення автомобіля.
   * 1. Сценарій “Видалення користувача”

Передумова

Адміністратор відкрив панель.

Основний сценарій:

1. Адміністратор відкриває вкладку з користувачами.
2. Адміністратор обирає користувача зі списку.
3. Адміністратор натискає кнопку “Delete”.
4. Програма просить підтвердження дії.
5. Адміністратор підтверджує дію.
6. Програма видаляє обраного користувача зі списку.

Додатковий сценарій

1. Адміністратор відкриває вкладку з користувачами.
2. Адміністратор обирає користувача зі списку.
3. Адміністратор натискає кнопку “Delete”.
4. Програма просить підтвердження дії.
5. Адміністратор cкасовує дію.
6. Програма скасовує операцію видалення користувача.
   * 1. Сценарій “Видалення заяви”

Передумова

Адміністратор відкрив панель.

Основний сценарій:

1. Адміністратор відкриває вкладку з заявами.
2. Адміністратор обирає заяву зі списку.
3. Адміністратор натискає кнопку “Delete”.
4. Програма просить підтвердження дії.
5. Адміністратор підтверджує дію.
6. Програма видаляє обрану заяву зі списку.

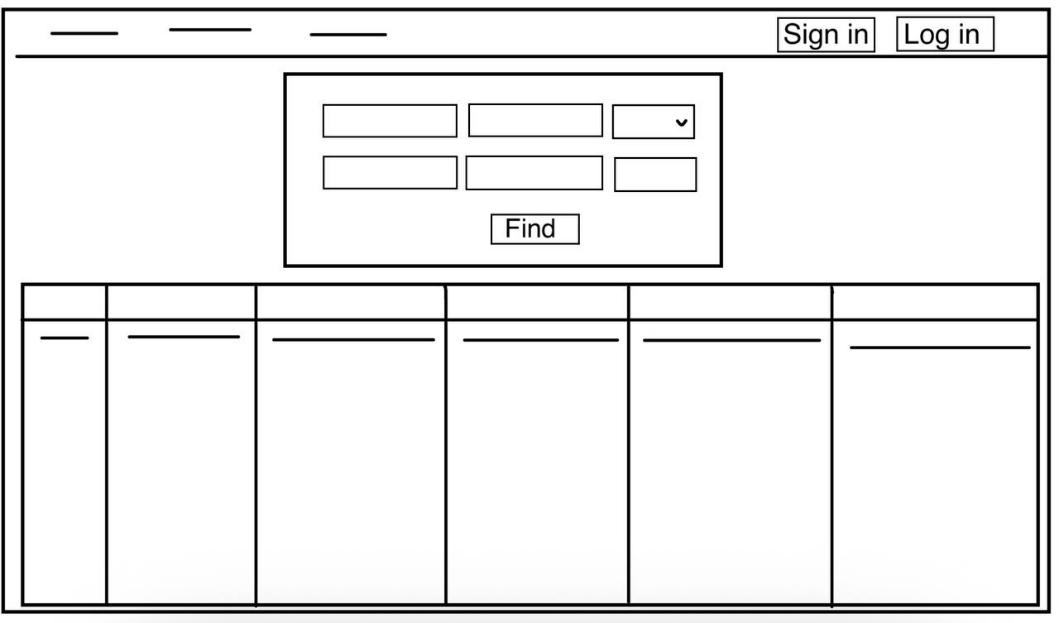
Додатковий сценарій

1. Адміністратор відкриває вкладку з заявами.
2. Адміністратор обирає заяву зі списку.
3. Адміністратор натискає кнопку “Delete”.
4. Програма просить підтвердження дії.
5. Адміністратор скасовує дію.
6. Програма скасовує операцію видалення заяви.
   1. Опис функцій
      1. Функція “Пошук автомобіля”

Панель пошуку автомобіля розташовується на головному вікні програми (див. рис. 1.1) і містить такі поля вводу:

* “Brand” – містить марку автомобіля;
* “Model” – містить модель автомобіля;
* “Year” – числове поле, що відповідає за рік випуску автомобіля;
* “Condition” – випадаючий список зі значеннями “All”, “New” та “Used”;
* “Minimum price” – числове поле;
* “Maximum price” – числове поле;

Під усіма текстовими полями знаходиться кнопка “Find”.

  
Рисунок 1.1 – Головне вікно програми

Якщо поле вводу залишено порожнім, відповідна умова не враховується під час пошуку. Тобто заповнювати всі поля не обов’язково — користувач може здійснювати пошук лише за тими параметрами, які його цікавлять.

Поля “Year”, “Minimum Price” та “Maximum Price” мають приймати лише числові дані, в іншому варіанті – вони не повинні бути застосованим для пошуку.

Результати пошуку відображаються у вигляді списку автомобілів з базовою інформацією: марка, модель, рік, стан, ціна.

Якщо жодного автомобіля не знайдено, замість списку автомобілів користувач побачить повідомлення: «No cars found matching the criteria».

* + 1. Функція “Реєстрація нового користувача”

Вікно реєстрації нового користувача це окреме вікно (див. рис. 1.2), яке відкривається при натисканні на кнопку “Sign up”. Форма містить такі поля:

* “E-mail” – електрона пошта користувача, буде використовуватися як логін;
* “Password” – пароль;
* “Brands” – список брендів, які користувач бажає придбати. Ці дані використовуються у функції автоматизованого пошуку;
* “Models” – список моделей, які користувач хоче купити. Ці дані використовуються у функції автоматизованого пошуку;
* “Minimum price” – мінімальна вартість автомобіля. Значення використовується у функції автоматизованого пошуку;
* Зображення, що містить текст, знімок екрана, монітор, число

  Вміст, створений ШІ, може бути неправильним.“Maximum price” – максимальна вартість автомобіля. Значення використовується у функції автоматизованого пошуку.

Рисунок 1.2 – Вікно для реєстрації нового користувача

Після натискання кнопки “Sign up”, програма перевіряє коректність введених даних:

* Значення в полі “E-mail” повинно відповідати формату: user@domain.com, де доменна частина складається частин, розділених крапкою, та не закінчується крапкою. Кожен обліковий запис має унікальну електронну адресу, тому під час реєстрації програма перевіряє список користувачів, щоб переконатися, що введена адреса ще не використовується.
* Значення в полі “Password” повинно складатися з щонайменше 8-ми символів;
* Значення в полях “Minimum price” та “Maximum price” повинні бути невід’ємними цілими числами. Крім того, “Maximum price” не може бути меншим за “Minimum price”;

При заповнені форми, користувач обов’язково повинен вказати:

* Поля “E-mail” та “Password” для створення облікового запису;
* Поля “Minimum price” та “Maximum price” для роботи функції автоматизованого пошуку.

У разі успішної реєстрації, користувача повертає на головне вікно програми, а правому верхньому кутку з’являється введена пошта.

* + 1. Функція “Авторизація користувача”

Вікно авторизації користувача це окреме вікно (див. рис. 1.3), яке відкривається при натисканні на кнопку “Log in”. Форма містить такі поля:

* “E-mail” – електронна пошта, вказана користувачем при реєстрації;
* “Password” – пароль;

Зображення, що містить текст, знімок екрана, схема, Прямокутник

Вміст, створений ШІ, може бути неправильним.

Рисунок 1.3 – Вікно для авторизації користувача

Для входу до системи користувач обов’язково повинен заповнити обидва поля.

Після натискання кнопки “Log in”, програма перевіряє наявність користувача у списку зареєстрованих. Можливі такі результати:

* Програма знайшла користувача – відкривається головне вікно програми з даними користувача (наприклад, логін, подані заяви на викуп);
* Програма не знаходить заданого користувача – програма повідомляє користувача про неправильно введену електрону пошту або пароль. Програма не закриває вікно авторизації;

Якщо жодне з полів не заповнене, програма повідомить користувача про необхідність заповнення полів для логіну та паролю.

* + 1. Функція “Автоматизований пошук автомобілів”

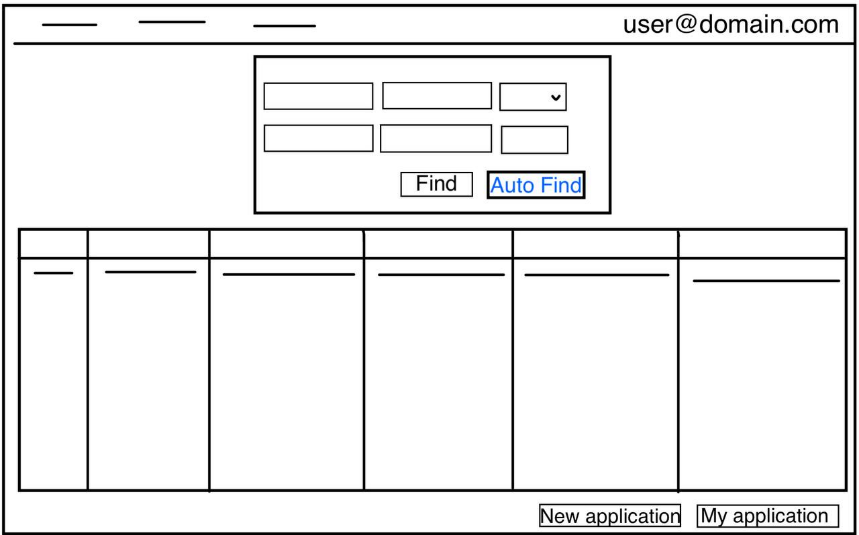
Кнопка “Auto Find” для автоматизованого пошуку з’являється на головному вікні програми після реєстрації або входу користувача в особистий кабінет (див. рис. 1.4).

Рисунок 1.4 – Головне вікно програми з кнопкою для автоматизованого пошуку автомобілів.

При натисканні на кнопку “Auto Find” виконується пошук автомобілів за параметрами, вказаними авторизованим користувачем під час реєстрації.

Результати пошуку відображаються у вигляді списку автомобілів з базовою інформацією: марка, модель, рік, стан, ціна.

Якщо жодного автомобіля не знайдено, замість списку автомобілів користувач побачить повідомлення: «No cars found matching the criteria».

* + 1. Функція “Подання заяви на викуп”

Кнопка “New Application” для подання заяви на викуп автомобілів, з’являється на головному вікні програми після реєстрації або входу користувача в особистий кабінет (див. рис. 1.4).

Для формування заяви користувач повинен обрати щонайменше один автомобіль.

При натисканні на кнопку “New Application”:

* Програма перевіряє чи користувач здійснив вхід в систему. Якщо ні – програма сповіщає користувача про необхідність входу;
* Програма формує заяву з обраними автомобілями та додає до неї унікальний ID;
* Програма сповіщає користувача про успішне формування заяви та показує ID цієї заяви;

Якщо користувач не обрав жодного авто, програма сповістить користувача про необхідність вибору хоча б одного авто для формування заяви.

* + 1. Функція “Перегляд надісланих заяв на викуп”

Кнопка “My Application” для перегляду надісланих заяви на викуп автомобілів, з’являється на головному вікні програми після реєстрації або входу користувача в особистий кабінет (див. рис. 1.4).

При натисканні на кнопку “My Application”

* Програма перевіряє чи користувач здійснив вхід в систему. Якщо ні – програма сповіщає користувача про необхідність входу;
* Зображення, що містить текст, знімок екрана, схема, ряд

  Вміст, створений ШІ, може бути неправильним.Програма відображає список заяв, надісланих користувачем (див. рис. 1.5);

Рисунок 1.5 – Відображення списку надісланих заяв на викуп

Якщо користувач ще не відправляв жодної заяви, програма сповіщає користувача, що надісланих заяв немає.

* + 1. Функція “Авторизація адміністратора”

Для доступу до панелі редагування даних, адміністратор використовує спеціальну комбінацію логіну та пароля (наприклад: admin / adminAutoMatcher).

Якщо хоча б один символ введено некоректно, система автоматично виконує стандартну функцію авторизації користувача.

У разі успішного введення адміністративних облікових даних, програма відкриває панель адміністратора (див. рис. 1.6).

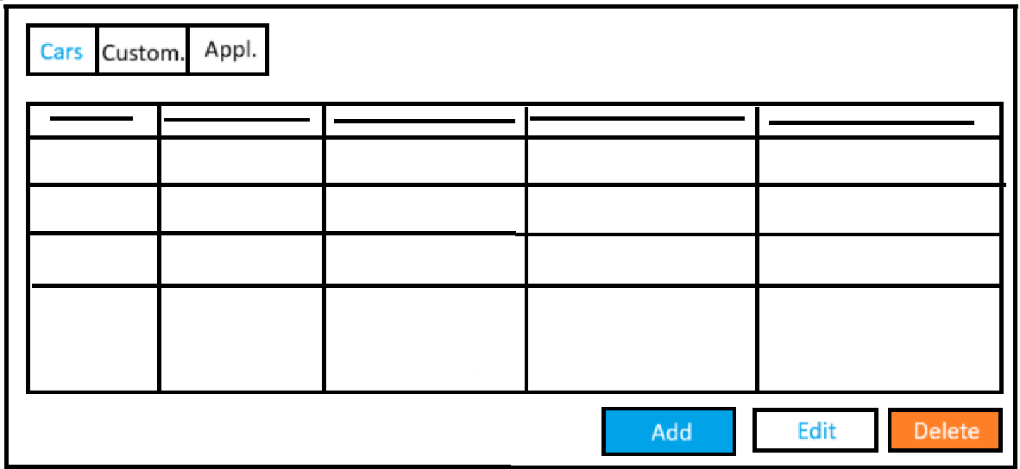


Рисунок 1.6 – Панель адміністратор системи при роботі з автомобілями

* + 1. Функція “Додавання автомобіля”

Зображення, що містить текст, знімок екрана, Шрифт, число

Вміст, створений ШІ, може бути неправильним.Для додавання нового автомобіля використовується вікно додавання автомобіля (див. рис. 1.7), яке відкривається після натискання кнопки “Add” під час роботи зі списком автомобілів.

Рисунок 1.7 – Вікно додавання нового автомобіля до списку

Вікно додавання нового автомобіля містить наступні поля:

* “Brand” – марка автомобіля;
* “Model” – модель автомобіля;
* “Country” – країна походження автомобіля;
* “Year” – рік виготовлення автомобіля;
* “Price” – ціна;
* “Engine” – тип двигуна;
* “HP” – числове значення потужності двигуна автомобіля, рахується в кінських силах;
* “Fuel” – тип пального;
* “Condition” – випадаючий список зі значеннями “New” та “Used”;

Значеннями полів “Brand”, “Model”, “Country”, “Engine”, “Fuel” повинні бути непорожніми рядками.

У полях “HP”, “Year”, “Price” повинні бути лише числа більше нуля. Крім того, значення поля “Year” повинно бути не більше ніж поточний рік.

При натисканні на кнопку “Add car” програма здійснює перевірку введених даних. Якщо хоча б одне поле заповнено некоректно, програма сповіщає про помилку у цьому полі.

При успішній перевірці даних, програма додає автомобіль, що містить введені дані, у список існуючих автомобілів.

* + 1. Функція “Редагування даних існуючого автомобіля”

Для редагування даних вже доданого автомобіля використовується вікно редагування автомобіля (див. рис. 1.8). Це вікно відкривається після натискання кнопки “Edit”, якщо під час роботи зі списком автомобілів було обрано конкретний автомобіль.

Зображення, що містить текст, знімок екрана, Шрифт, число

Вміст, створений ШІ, може бути неправильним.Рисунок 1.8 – Вікно редагування існуючого автомобіля

Вікно редагування існуючого автомобіля містить наступні поля:

* “Brand” – марка автомобіля;
* “Model” – модель автомобіля;
* “Country” – країна походження;
* “Year” – рік виготовлення;
* “Price” – вартість;
* “Engine” – тип двигуна;
* “HP” – потужність двигуна (у кінських силах);
* “Fuel” – тип пального;
* “Condition” – випадаючий список зі значеннями “New” та “Used”;

Вимоги до введення даних:

* Поля “Brand”, “Model”, “Country”, “Engine”, “Fuel” повинні містити непорожні текстові дані;
* Поля “Brand”, “Model”, “Country”, “Engine”, “Fuel” повинні містити цілі числа більше нуля. Крім того, значення поля “Year” повинно бути не більше ніж поточний рік;

При натисканні на кнопку “Edit” програма здійснює перевірку введених даних. Якщо хоча б одне поле заповнено некоректно, програма сповіщає про помилку у цьому полі.

Для збереження змін обов’язково треба натиснути на кнопку “Edit” у вікні редагування даних, інакше дані не будуть збережено.

Після успішного редагування програма зберігає оновлені дані автомобіля, оновлює відображення списку автомобілів.

* + 1. Функція “Видалення автомобіля”

Видалення автомобіля відбувається в панелі адміністратора при роботі з автомобілями (див рис. 1.6).

Видалення можливе лише за умови, що адміністратор обрав автомобіль зі списку. При натисканні на кнопку “Delete”:

* Програма просить підтвердження дії видалення обраного автомобіля, відкриваючи вікно підтвердження дії видалення з даними автомобіля (див. рис. 1.9);
* Адміністратор має або підтвердити видалення, або відмінити дію;

Якщо адміністратор підтвердив операцію видалення, обраний автомобіль видаляється зі списку.

Рисунок 1.9 – Вікно підтвердження видалення даних

* + 1. Функція “Видалення користувача”

Видалення користувача відбувається в панелі адміністратора при роботі з користувачами (див рис. 1.10).

Видалення можливе лише за умови, що адміністратор обрав користувача зі списку. При натисканні на кнопку “Delete”:

* Програма просить підтвердження дії видалення обраного користувача, відкриваючи вікно підтвердження дії видалення з даними користувача (див. рис. 1.9);
* Адміністратор має або підтвердити видалення, або відмінити дію;

Зображення, що містить знімок екрана, ряд, Прямокутник, Паралель

Вміст, створений ШІ, може бути неправильним.Якщо адміністратор підтвердив операцію видалення, обраний користувач видаляється зі списку.

Рисунок 1.10 – Панель адміністратора системи при роботі з користувачами

* + 1. Функція “Видалення заяви на викуп”

Видалення заяви на викуп відбувається в панелі адміністратора при роботі з користувачами (див рис. 1.11).

Видалення можливе лише за умови, що адміністратор обрав заяву зі списку. При натисканні на кнопку “Delete”:

* Програма просить підтвердження дії видалення обраної заяви, відкриваючи вікно підтвердження дії видалення з даними заяви (див. рис. 1.9);
* Адміністратор має або підтвердити видалення, або відмінити дію;

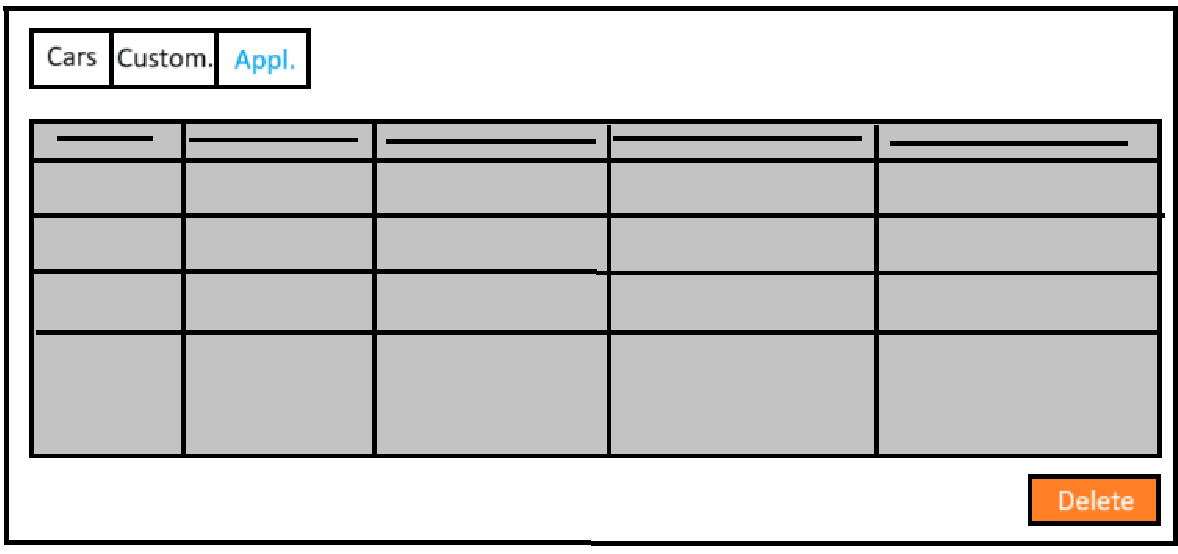
Якщо адміністратор підтвердив операцію видалення, обрана заява видаляється зі списку.

Рисунок 1.11 – Панель адміністратора системи при роботі з заявами

# ПРОЄКТУВАННЯ ПРОГРАМИ

* 1. Визначення архітектури проєкту

Проєкт реалізовано з використанням настільної архітектури, що передбачає графічний інтерфейс користувача (GUI), створений на базі фреймворку Windows Forms.

Для забезпечення підтримуваності, розширюваності та повторного використання логіки, проєкт структуровано по модулях (папках), зокрема:

* “Forms” – містить усі форми програми, які використовуються під час роботи користувача з інтерфейсом. Тут реалізується взаємодія з бізнес-логікою через обробники подій, кнопки, текстові поля тощо;
* “Models” – містить основні моделі проєкту. Ці класи відповідають за представлення структури даних і забезпечують базову функціональність, необхідну для логіки програми;
* “Project” – містить головний фасадний клас, який реалізує усю бізнес**-**логіку**;**
* “StorageData” – містить клас, який використовується для формування тестових даних;
  1. Визначення основних класів програми

Для реалізації системи, що працює з автомобілями, користувачами та заявами, у програмі були створені відповідні класи, кожен з яких виконує чітко визначену функцію. Нижче наведено короткий опис основних класів:

* “Car” – клас для опису автомобіля. Містить інформацію про марку, модель, *технічні характеристики*, стан та ціну;
* “Features” – допоміжний клас для збереження технічних характеристик автомобіля (тип двигуна, потужність двигуна, тип пального);
* “Customer” – клас, що представляє клієнта системи та його обліковий запис. Зберігає електрону пошту, пароль, бюджет, бажані марки та моделі;
* “CarApplication” – клас, що зберігає заяви клієнтів на вибрані автомобілі. Містить список обраних автомобілів, електрону пошту клієнту та унікальний ідентифікатор заяви;

У програмі реалізовані спеціальні класи-колекції, які відповідають за зберігання та управління списками об’єктів основних типів: автомобілів, клієнтів та заяв. Вони інкапсулюють логіку роботи з відповідними колекціями елементів та надають базові операції для взаємодії з ними:

* “CarDataBase” – клас, що відповідає за зберігання колекції автомобілів. Містить базові методи та метод фільтрування за певними параметрами ( рік, ціна, марка тощо);
* “CustomerDataBase” – клас для збереження зареєстрованих клієнтів. Має базові методи та метод перевірки наявності зареєстрованого клієнта за поштою та паролем;
* “ApplicationDataBase” – клас для роботи із заявами клієнтів за допомогою базових методів;

Фасадний клас ProjectModel, який виконує роль центральної системи координації роботи всіх класів-колекцій, містить інформацію про поточного клієнта та надає методи для фільтрації і пошуку заявок. Крім того, цей клас відповідає за завантаження та збереження даних у форматі JSON із використанням JSON-серіалізації.

Зображення, що містить текст, схема, ряд, Шрифт

Вміст, створений ШІ, може бути неправильним.Повну діаграму класів можна побачити на рис. 2.1

Рисунок 2.1 – Діаграма класів

1. ІНСТРУКЦІЯ КОРИСТУВАЧА
   1. Встановлення програми

Перед завантаженням застосунку AutoMatcher, користувач повинен завантажити середовище виконання платформи .NET - .NET 8.0 Runtime.