# *Високорівневий опис реалізації:*

**Архітектурна концепція:**

* Створення GUI:

Розробка графічного інтерфейсу користувача (GUI) за допомогою інструменту Windows Forms, який забезпечить зручну взаємодію користувача з програмою.

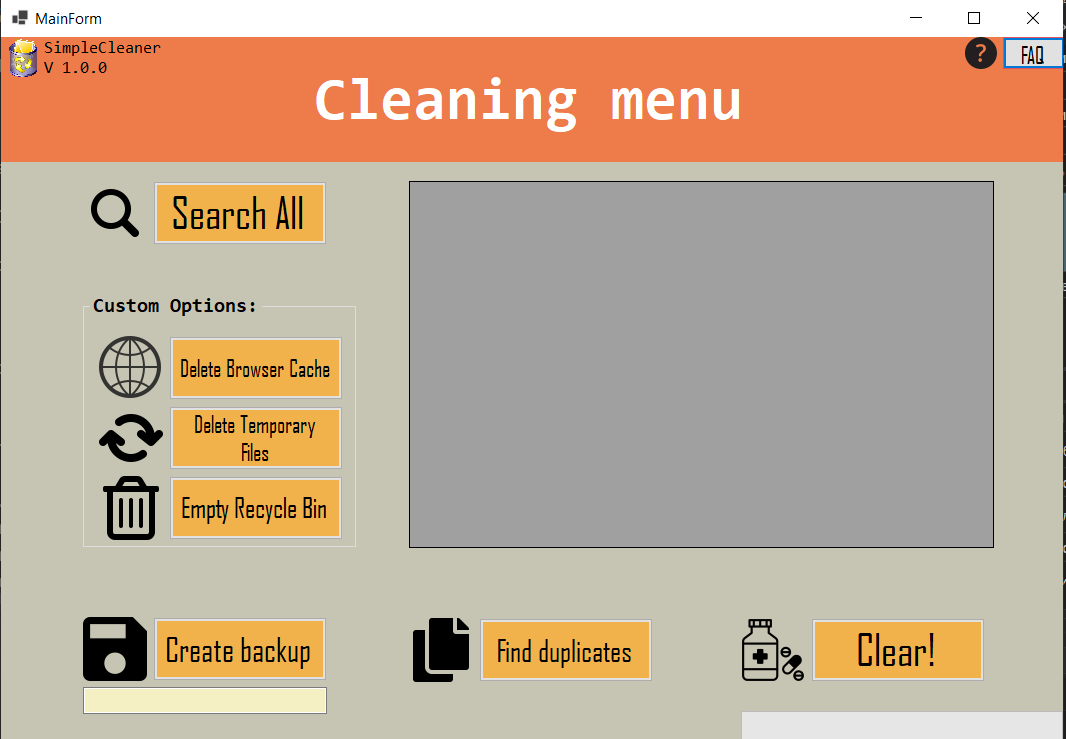
* Розробка програмної логіки:

Розробка функціональності програми, що включає сканування системи для виявлення непотрібних файлів, видалення тимчасових файлів, очищення кешу, видалення тимчасових Internet файлів, видалення файлів з кошика, пошук та видалення дублікатів файлів.

* Інтеграція функціоналу з GUI:

Побудова взаємодії між графічним інтерфейсом та програмною логікою для обробки користувацьких взаємодій і виконання відповідних дій.

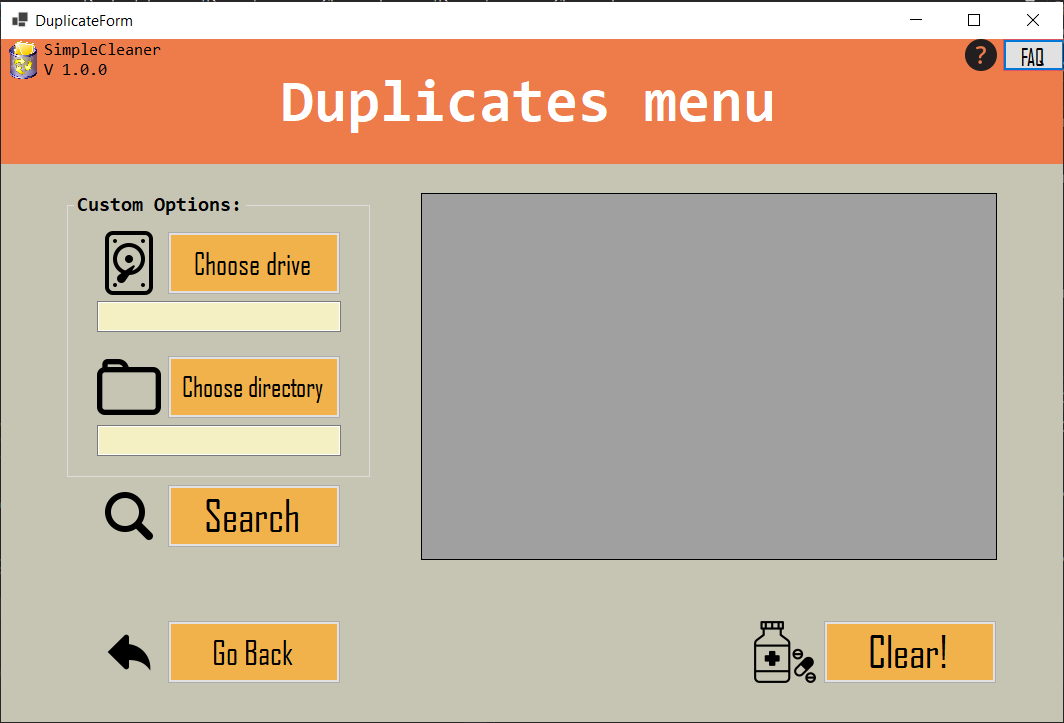
**Приблизний вигляд GUI з деякими реалізованими функціями:**



*Рис.1 Вигляд головного меню програми, а саме форми CleaningForm*

Як бачимо, користувач має можливість провести сканування всіх “сміттєвих” файлів, або видалити окремі їх типи. Також, у боковому вікні (DataGridView) виводиться інформація про дані файли. Також у користувача присутня можливість створити бекап файлів перед їх видаленням. Ось короткий опис функціоналу кожної клавіші:

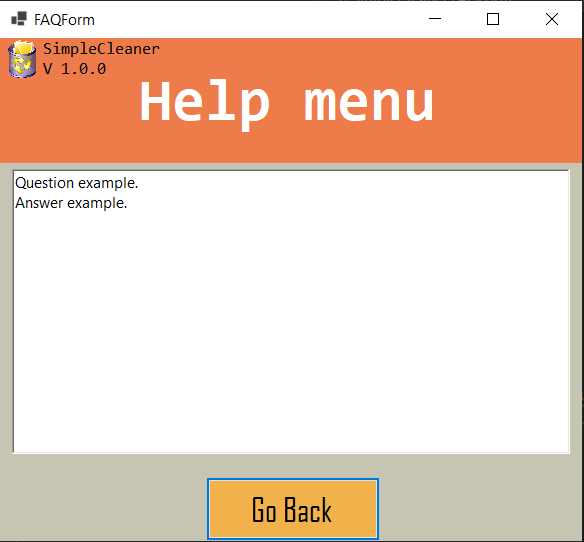
* Клавіша “Search All” проводить сканування усіх сміттєвих файлів, та виводить інформацію про них.
* Клавіші у розділі “Custom Options” проводять сканування сміттєвих файлів за певними критеріями, такими як: очищення кешу браузерів, очищення тимчасових файлів Windows та очищення кошика.
* Клавіша “Create backup” дозволяє користувачу зробити бекап файлів перед їх видаленням.
* Клавіша “Find duplicates” перенаправляє користувача у іншу форму, яка буде відповідати за роботу з дублікатами.
* Клавіша “Clear!” проводить очистку вибраних файлів.



*Рис.2 Вигляд другорядного меню програми, а саме форми DuplicateForm*

Як бачимо, у цьому вікні користувач має можливість створити дублікати вибраних ним файлів. Також є клавіша повернення назад, тобто у головне меню програми. Форми програми в цілому схожі одна на одну, але притримуються однієї стилістики, виконуючи різний функціонал. Ось короткий опис функціоналу кожної клавіші:

* Клавіша “Choose drive” дозволяє користувачу провести сканування на дублікати у всьому логічному диску,
* Клавіша “Choose directory” проводить сканування лише у певній директорії.
* Клавіша “Search” проводить сканування на наявність дублікатів, та виводить інформацію про них.
* Клавіша “Clear!” видаляє дублікати файлів.

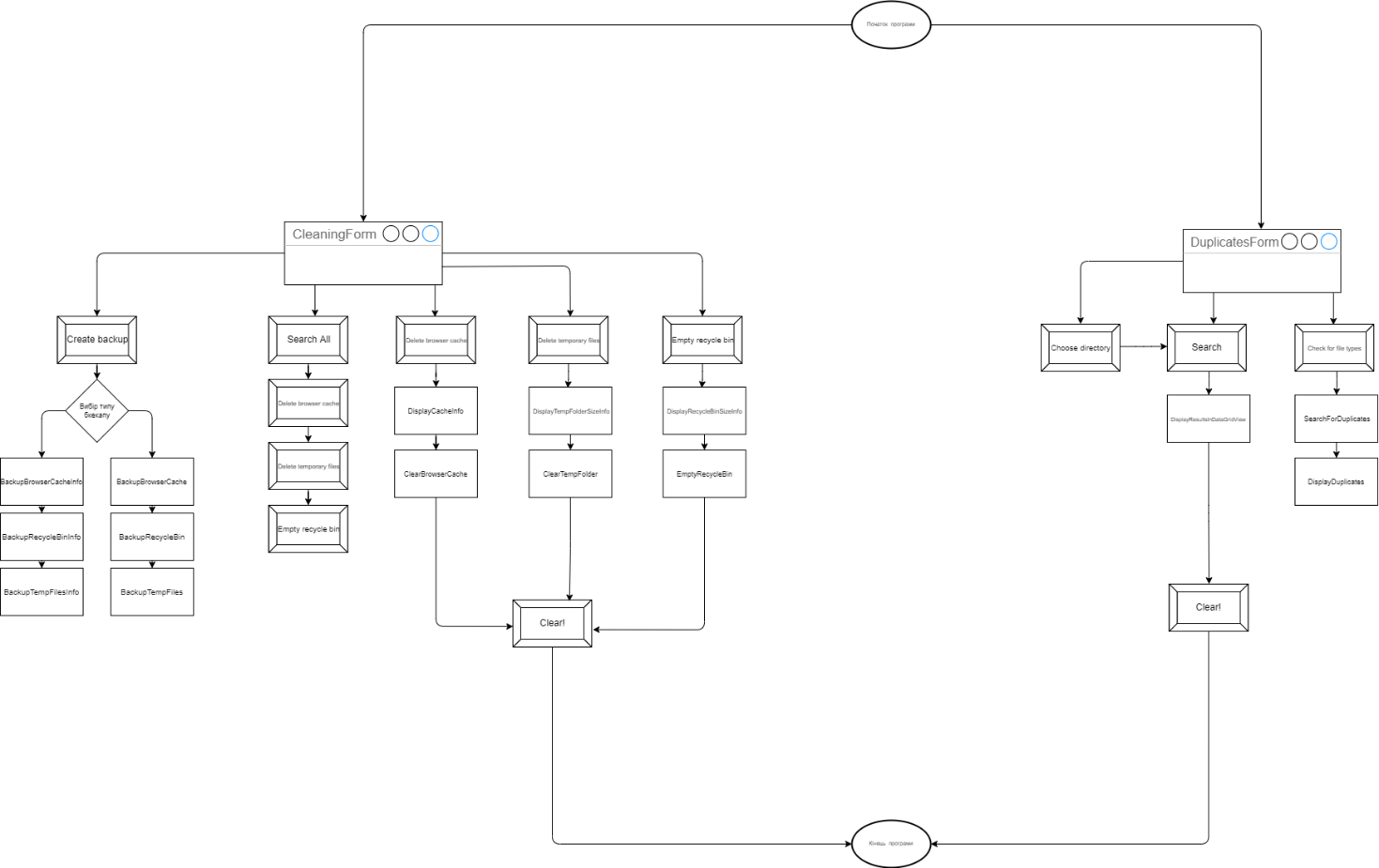


*Рис.3 Вигляд другорядного меню програми, а саме форми FAQForm*

У цьому вікні відображається інформація про програму, та відповіді на найчастіші питання. Було прийнято рішення не ускладнювати дане завдання, і реалізувати вивід даної інформації у простому RichTextBox.

# *Структура програмного рішення:*

* **Структурна схема програми:**

****

*Рис.4 Структурна схема програми*

В даній структурній схемі програми описаний весь функціонал програми, та дії, які буде виконувати кожен елемент програми.

* **Вибір засобів для розробки:**

Так як було зазначено раніше, Для розробки GUI використовується Windows Forms. Програмна логіка реалізована за допомогою мови програмування C# та вбудованих засобів .NET для роботи з файлами та директоріями.

Загалом вся логіка програми реалізована в окремих функціях, які викликаються під час натискання певних клавіш користувачем, після чого результати виконання даних функцій виводяться в таких елементах як DataGridView або MessageBox.

* **Архітектура класів:**

В кожній формі буде реалізовано свій клас, на даному етапі розробки програма має такі класи:

public partial class CleaningForm : Form {}

public partial class DuplicateForm : Form {}

public partial class FAQForm : Form {}

Основна ідея у використанні класів у Windows Forms полягає в тому, щоб кожен клас-форма мав свій власний набір функціональності, властивостей та подій, які відповідають конкретному елементу інтерфейсу користувача вашого додатку.

# *Перша спроба реалізації каркасу програмного рішення:*

Приблизний вигляд програмного рішення вже є, в нього можуть вноситись деякі корективи. Тепер необхідно провести огляд реалізації всього задуманого функціоналу. Весь код знаходиться у трьох файлах, а саме: DuplicateForm.cs, CleaningForm.cs, FAQForm.cs. Весь код відповідно розбитий за своїм функціоналом. Для кращої читабельності, та зручності, код також розбитий на функції. В свою чергу, функції викликаються під час натискання користувачем на певні клавіші.

* **Ось приблизний вигляд виклику функції:**

private async void RecycleBinButton\_Click(object sender, EventArgs e)

{

ProgressBar.Value = 0;

await SimulateLoadingAsync(1100);

ResultGridView.Rows.Clear();

ResultGridView.ColumnCount = 2;

ResultGridView.Columns[0].Name = "Drive name:";

ResultGridView.Columns[1].Name = "Recycle Bin Size (MB):";

ResultGridView.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

DisplayRecycleBinSizeInfo();

EmptyRecycleBin();

}

Логіка цього коду досить проста, налаштовуємо DataGridView для виводу певної інформації, а потім по черзі викликаємо дві функції, DisplayRecycleBinSizeInfo(); та EmptyRecycleBin(); відповідно для відображення та очищення файлів з кошика. У функції EmptyRecycleBin(); передбачений окремий MessageBox з підтвердженням дій користувача, якщо натискання на клавішу було випадковим.

* **Результат натискання даної клавіші на практиці:**

