|  |  |  |  |
| --- | --- | --- | --- |
| **Laboratorium 4**  **Sprawozdanie z realizacji laboratorium** | | | |
| **Temat:** Praktyczne wykorzystanie ORM + praca z GUI | **Nr Albumu:** 028487 | **Grupa/zespół:** GL01 | **Rok/semestr:** III / 6 |
| **Wykonał:**  Oleksii Hudzishevskyi | **Data wykonania:**  15/05/2023 | | **Data oddania:**  15/05/2023 |
| **Ocena:** | | **Podpis prowadzącego:** |

# Spis treści

[1. Spis treści 1](#_Toc133911124)

[2. Cel ćwiczenia 2](#_Toc133911125)

[3. Wymagania znajomości zagadnień 2](#_Toc133911126)

[4. Literatura, materiały dydaktyczne 2](#_Toc133911127)

[5. Wiadomości teoretyczne. 3](#_Toc133911128)

[6. Opracowanie sprawozdania 3](#_Toc133911129)

[7. Wnioski 5](#_Toc133911130)

[8. Bibliografia 5](#_Toc133911131)

[9. Spis ilustracji 5](#_Toc133911132)

[10. Spis snippetów 5](#_Toc133911133)

# Cel ćwiczenia

Laboratorium ma na celu poszerzenie wiedzy praktycznej w zakresie wykorzystywania narzędzi ORM w trakcie tworzenia aplikacji.

# Wymagania znajomości zagadnień

* Pisanie prostych aplikacji w C# lub innym obiektowym języku wysokiego poziomu
* Umiejętność tworzenia prostych aplikacji z wykorzystaniem GUI np. WinForms
* Umiejętność pracy na kolekcjach obiektów, proste zapytania LINQ.

# Literatura, materiały dydaktyczne

* <https://learn.microsoft.com/pl-pl/ef/core/managing-schemas/scaffolding/?tabs=dotnet-core-cli>
* <https://www.entityframeworktutorial.net/efcore/create-model-for-existing-database-in-ef-core.aspx>
* <https://www.learnentityframeworkcore.com/walkthroughs/existing-database>
* <https://www.mdatelier.pl/entity-framework-database-first/>
* <https://www.entityframeworktutorial.net/entityframework6/dbcontext.aspx>
* <https://learn.microsoft.com/pl-pl/ef/ef6/fundamentals/working-with-dbcontext>
* <https://learn.microsoft.com/pl-pl/dotnet/api/system.windows.forms.usercontrol?view=windowsdesktop-7.0>
* <https://www.c-sharpcorner.com/UploadFile/c5c6e2/creating-a-simple-user-control-and-adding-it-to-the-toolbox/>
* <https://www.c-sharpcorner.com/article/event-handling-in-net-using-C-Sharp/>
* <https://learn.microsoft.com/en-us/dotnet/standard/events/>
* <https://www.csharptutorial.net/csharp-tutorial/csharp-events/>
* <https://dotnetcorecentral.com/blog/delegates-and-events/>
* <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-access-objects-bound-to-windows-forms-datagridview-rows?view=netframeworkdesktop-4.8>
* <https://www.youtube.com/watch?v=GyLlpBZGsrE>

# Wiadomości teoretyczne

–

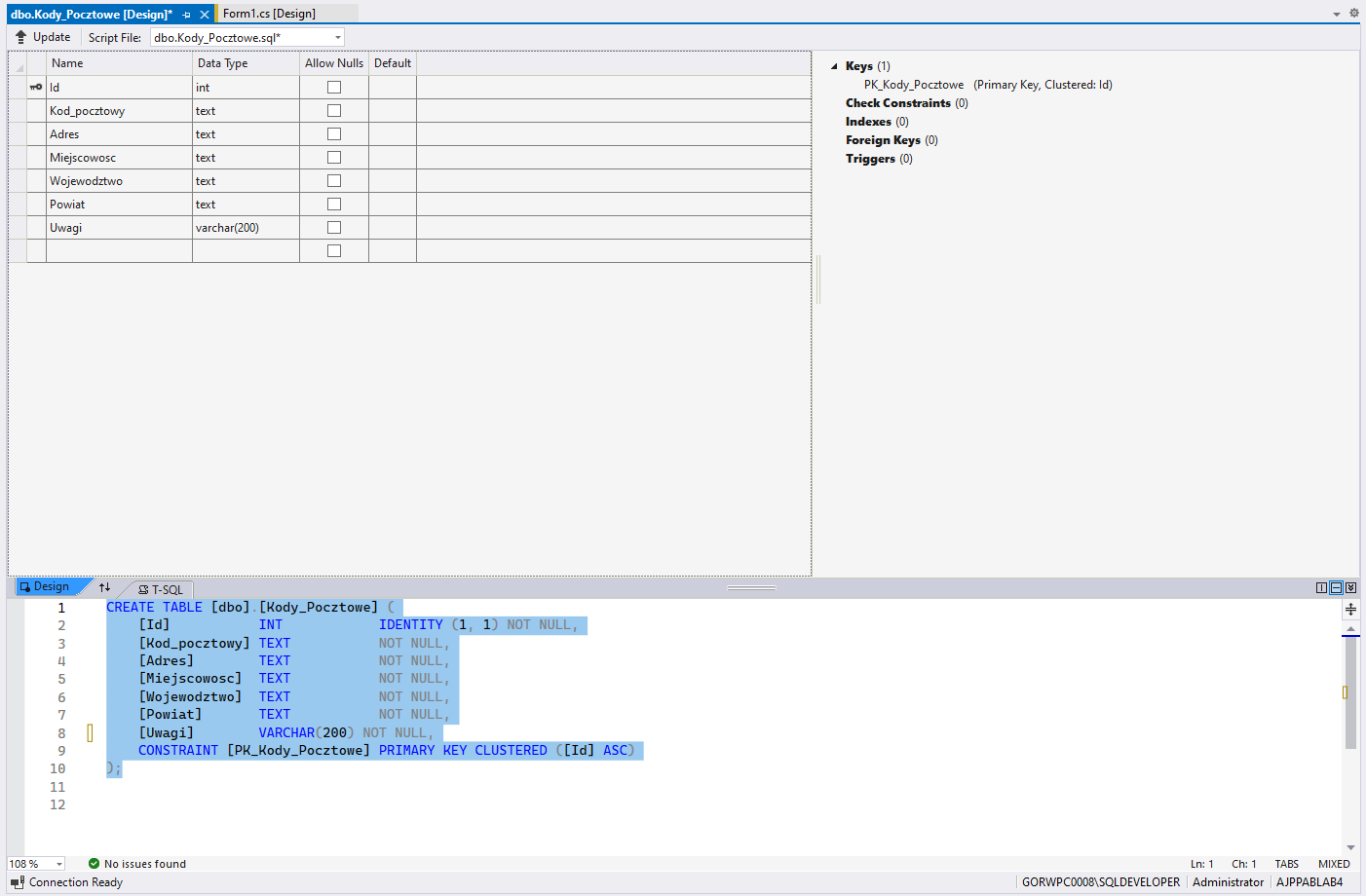
# Przebieg ćwiczenia praktycznego

Wykorzystując danych (tabelę) wraz z danymi z poprzedniego laboratorium, zmodyfikować tabelę tak by zawierała dodatkowo 2 pola: ID, Uwagi. Id ma być kluczem głównym w tabeli z opcją autoincreent a pole uwag polem Varchar(200). Dodać tabele lokalizacji (mogą to być sklepy, biura inne miejsca), tabela ta ma zawierać następujące pola: ID, Nazwa, Opis, oraz dodatkowo ID odnoszące się do tabeli kodów pocztowych (klucz obcy). Utworzyć projekt aplikacji WinForms, dołączyć wymagane nugety EF oraz wykorzystując polecenie Scaffold utworzyć context bazy danych wraz z modelami klas. Stworzyć kontrolkę użytkownika, która pozwalać będzie na wyświetlenie oraz edycję danych (zmiana lub kasowanie), oraz zagnieżdżony DataGridView zawierający listę wpisów powiązanych, kontrolka ta jako parametr konstruktora ma przyjmować obiekt Kodu Pocztowego. Do projektu dodać formularz wyszukiwania z poprzedniego laboratorium rozbudowany o możliwość wyszukiwania również po uwagach oraz stworzyć Datagridview zawierający wszystkie spełniające kryteria wyszukiwania obiekty. Pod DataGridView dodać przycisk do edycji, jego kliknięcie ma wczytać z górnego DataGridView zaznaczoną linię (o ile jest jakaś zaznaczona) a w przypadku gdy żadna linia nie została wybrana ma wyświetlić stosowny komunikat, a następnie dynamicznie stworzyć obiekt kontrolki zawierający dany wpis, na tej kontrolce można będzie zmienić dane (nazwa ulicy, pole uwag, skasować, dodać lub edytować wpisy z tabeli lokalizacji przypisanych do danego kodu pocztowego). Kliknięcie przycisku akcji w kontrolce (kasowanie, edycja dla kodu pocztowego) powinno wyświetlić komunikat o rezultacie operacji (OK lub Error wraz z podaniem przyczyny), zamknąć kontrolkę (usunąć ją z ekranu) oraz odświeżyć dane w głównym oknie programu. W przypadku dodawania, edycji i kasowania lokalizacji przypisanych do kodu (zagnieżdżony DataGridView na kontrolce), powinno to się odbywać bezpośrednio w kontrolce. Zadbać o komunikację pomiędzy kontrolką a głównym oknem aplikacji, tak by „wiedziało ono” o tym, że wprowadzono jakiekolwiek zmiany i że lista w głównym oknie programu powinna zostać odświeżona – w celu tym wykorzystać zdarzenia, po utworzeniu kontrolki należy przypisać w formularzu głównym zdarzenie zdefiniowane w tejże kontrolce. Przeładowanie listy na głównym oknie aplikacji ma się odbywać tylko i wyłącznie jeżeli w trakcie pracy użytkownik wprowadzi jakiekolwiek zmiany – kontrolka zarejestrowała fakt edycji i zakomunikowała to w stosownym zdarzeniu. Oczywiście wszystkie wprowadzane zmiany mają być wykonywane na bazie danych.

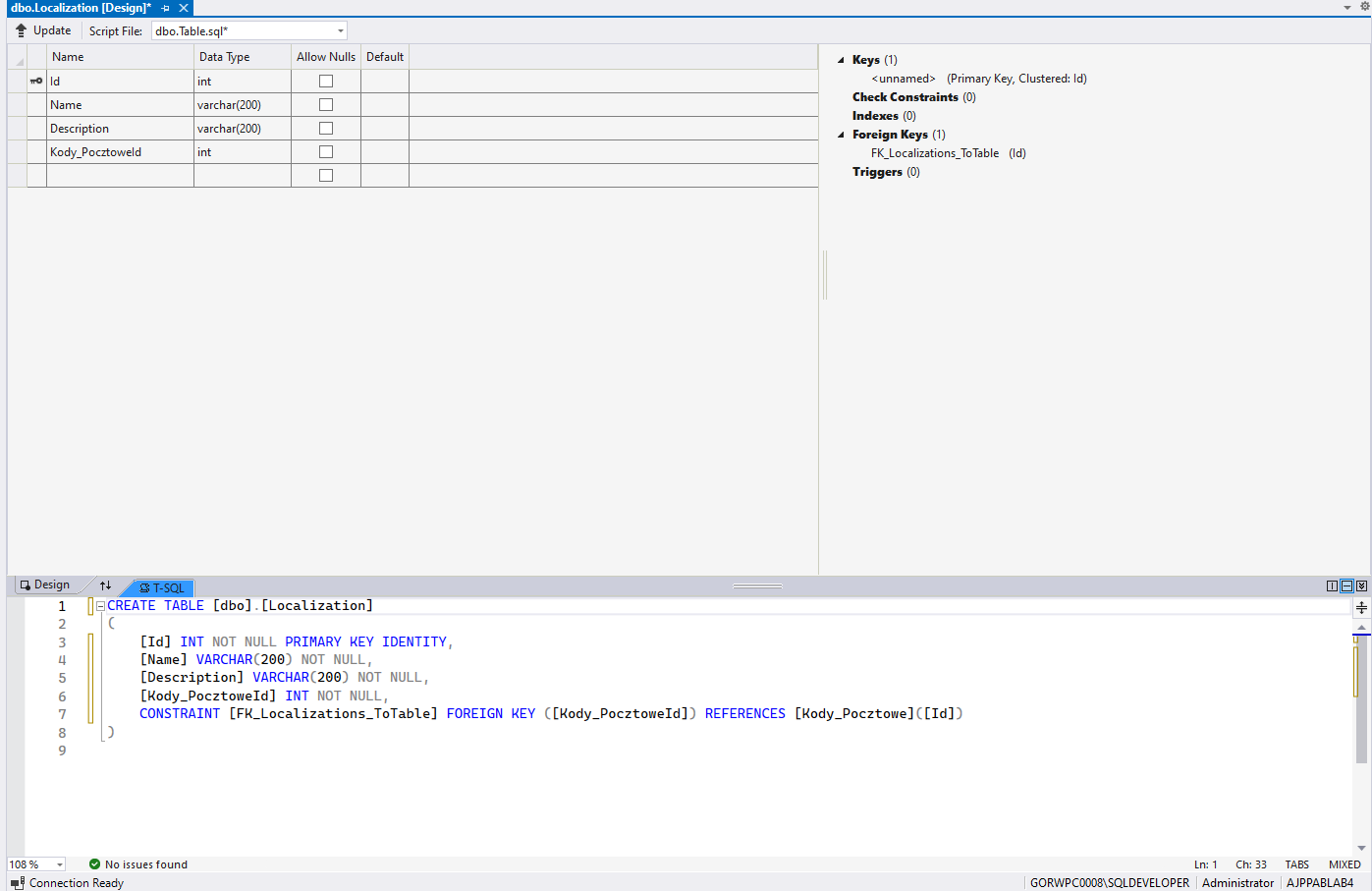
Ze względu na to, że opis interface może być niezrozumiały, wszelkie wątpliwości zostaną wyjaśnione w trakcie trwania laboratorium, na którym to, prowadzący omówi wygląd GUI oraz rozmieszczenie elementów - przy tablicy. Prowadzący przewiduje, iż wszystkie pytania oraz szczegóły implementacyjne zostaną omówione w trakcie trwania zajęć laboratoryjnych, osoby obecne w trakcie trwania zajęć będą mogły się dodatkowo skonsultować zdalnie. W przypadku osób nieobecnych na zajęciach które uprzednio nie zgłosiły i usprawiedliwiły faktu nieobecności wsparcie to nie będzie możliwe. Możliwość konsultacji - jedynie dla studentów mających dobre usprawiedliwienie swej nieobecności. W wyjątkowym wypadku, do uznania prowadzącego, usprawiedliwienie nieobecności może nastąpić po zajęciach, ale zostanie zaakceptowane jedynie wtedy gdy będzie naprawdę zasadne. Informuję również, iż przez wzgląd na notoryczne zaniżanie frekwencji i niestawianie się na zajęciach, poziom kolejnych zadań laboratoryjnych będzie zdecydowanie zawyżony, przy podniesionej punktacji oraz przy konsultacjach zarezerwowanych jedynie dla osób obecnych na zajęciach lub też posiadających stosowne usprawiedliwienie swej nieobecności. Do sprawozdania należy OBOWIĄZKOWO dołączyć spakowany projekt aplikacji, zawierający wszystkie kody źródłowe.

# Opracowanie sprawozdania

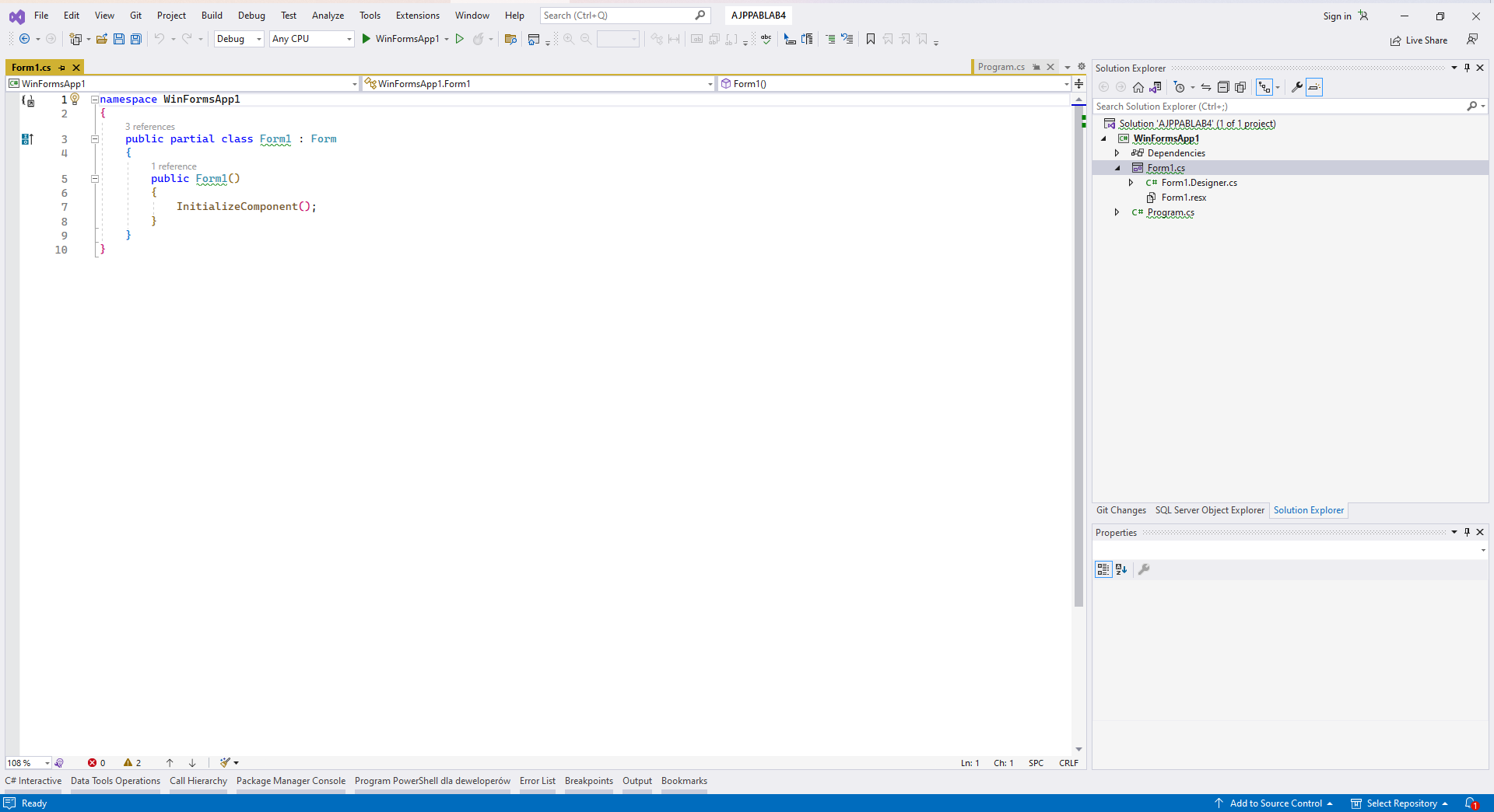
Wykorzystując danych (tabelę) wraz z danymi z poprzedniego laboratorium, zmodyfikować tabelę tak by zawierała dodatkowo 2 pola: ID, Uwagi. Id ma być kluczem głównym w tabeli z opcją autoincreent a pole uwag polem Varchar(200).



Dodać tabele lokalizacji (mogą to być sklepy, biura inne miejsca), tabela ta ma zawierać następujące pola: ID, Nazwa, Opis, oraz dodatkowo ID odnoszące się do tabeli kodów pocztowych (klucz obcy).

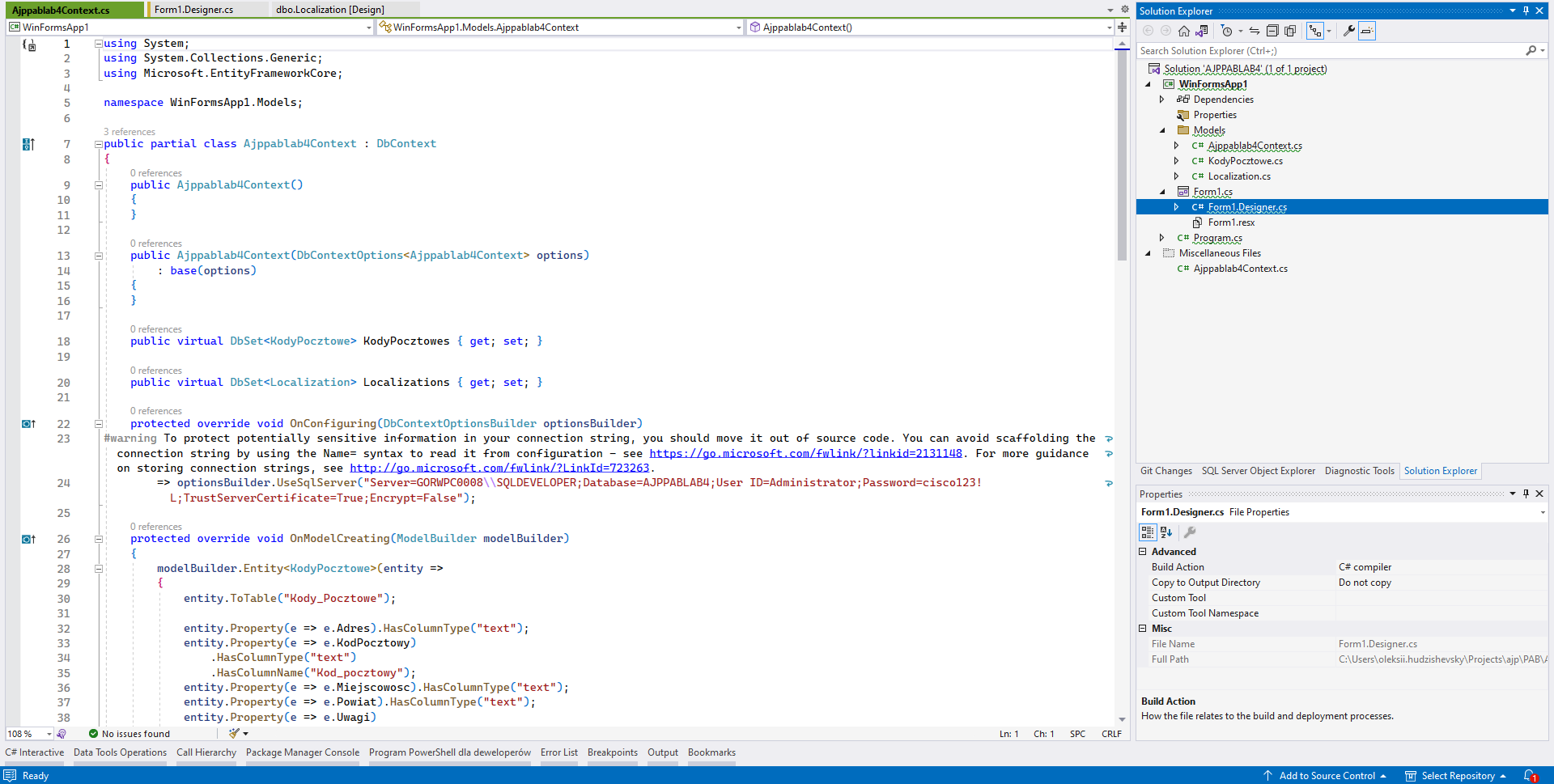


Utworzyć projekt aplikacji WinForms, dołączyć wymagane nugety EF oraz wykorzystując polecenie Scaffold utworzyć context bazy danych wraz z modelami klas.



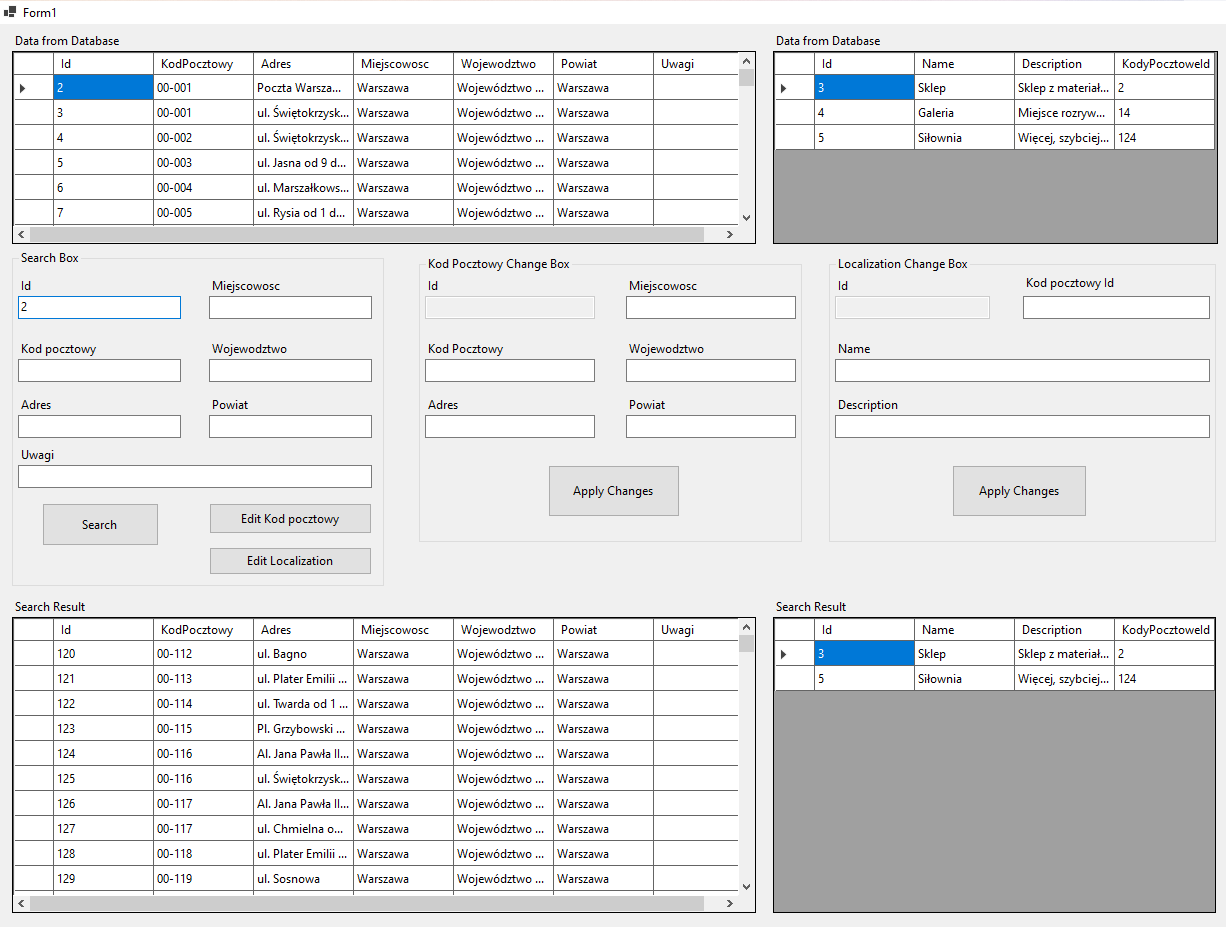
|  |
| --- |
| Install-Package Microsoft.EntityFrameworkCore  Install-Package Microsoft.EntityFrameworkCore.Relational  Install-Package Microsoft.EntityFrameworkCore.Abstractions  Install-Package Microsoft.EntityFrameworkCore.Design  Install-Package Microsoft.EntityFrameworkCore.Tools  Install-Package Microsoft.EntityFrameworkCore.Proxies  Install-Package Microsoft.EntityFrameworkCore.SqlServer |

|  |
| --- |
| Scaffold-DbContext "Server=GORWPC0008\SQLDEVELOPER;Database=AJPPABLAB4;User ID=Administrator;Password=cisco123!L;TrustServerCertificate=True;Encrypt=False" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models |



Stworzyć kontrolkę użytkownika, która pozwalać będzie na wyświetlenie oraz edycję danych (zmiana lub kasowanie), oraz zagnieżdżony DataGridView zawierający listę wpisów powiązanych, kontrolka ta jako parametr konstruktora ma przyjmować obiekt Kodu Pocztowego.

Do projektu dodać formularz wyszukiwania z poprzedniego laboratorium rozbudowany o możliwość wyszukiwania również po uwagach oraz stworzyć Datagridview zawierający wszystkie spełniające kryteria wyszukiwania obiekty.

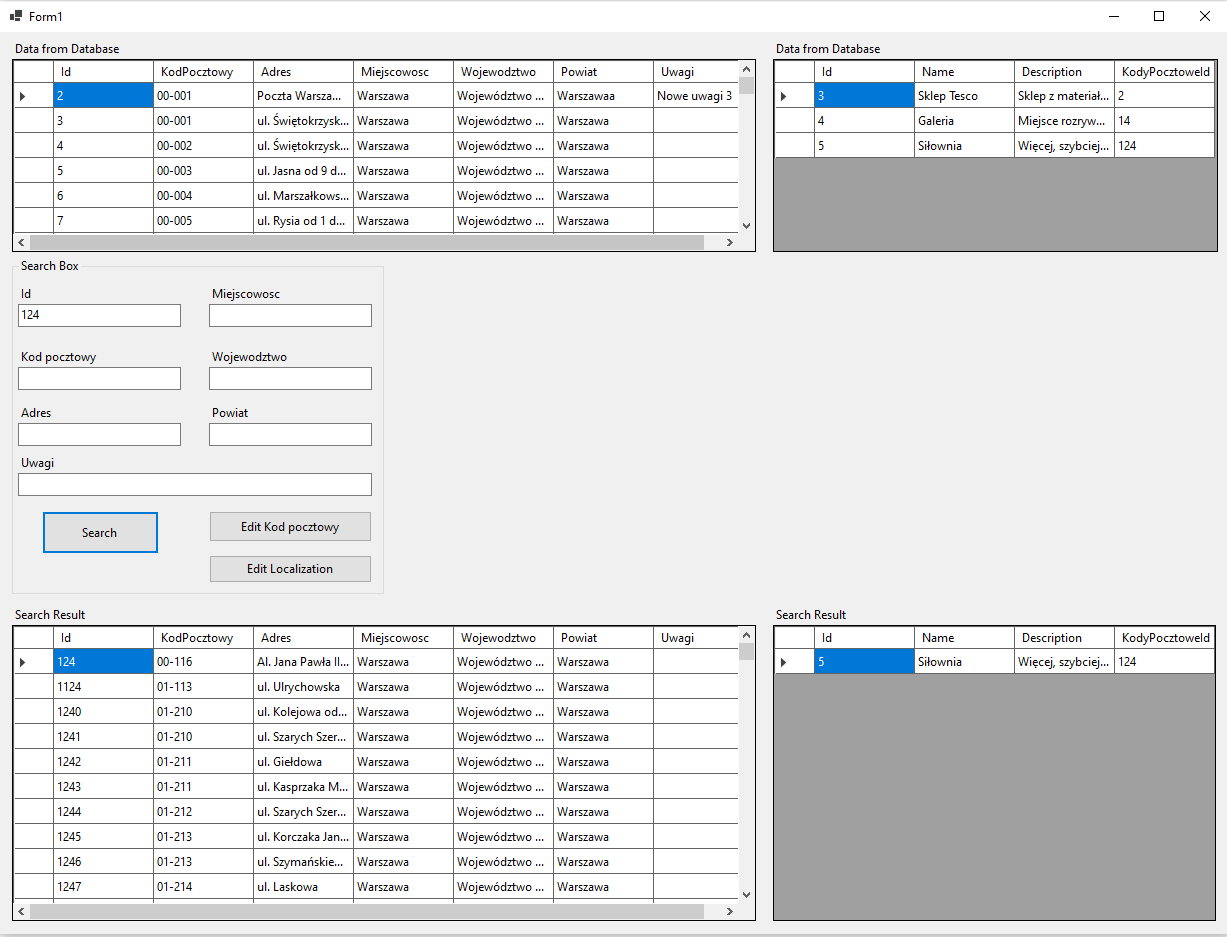


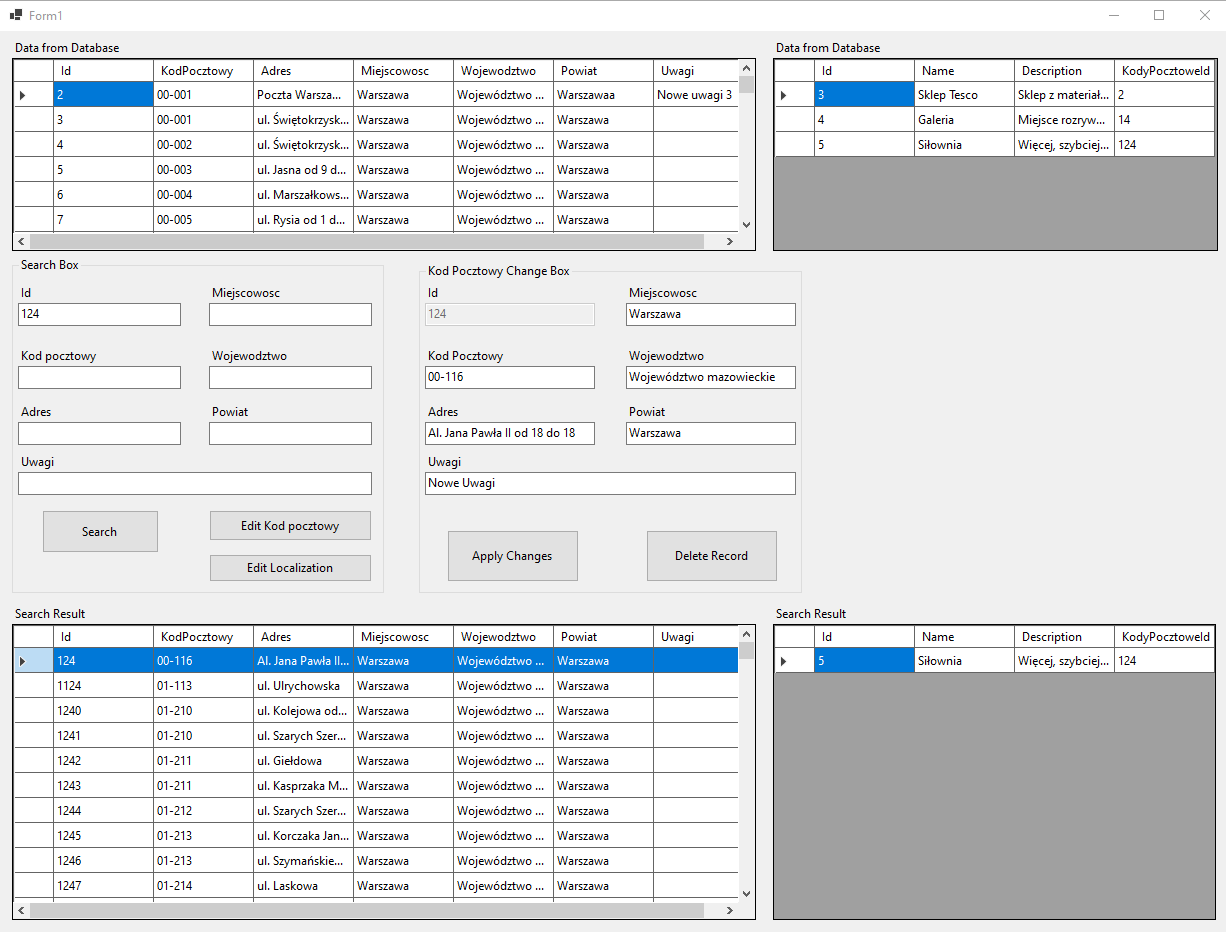
Pod DataGridView dodać przycisk do edycji, jego kliknięcie ma wczytać z górnego DataGridView zaznaczoną linię (o ile jest jakaś zaznaczona) a w przypadku gdy żadna linia nie została wybrana ma wyświetlić stosowny komunikat, a następnie dynamicznie stworzyć obiekt kontrolki zawierający dany wpis, na tej kontrolce można będzie zmienić dane (nazwa ulicy, pole uwag, skasować, dodać lub edytować wpisy z tabeli lokalizacji przypisanych do danego kodu pocztowego).

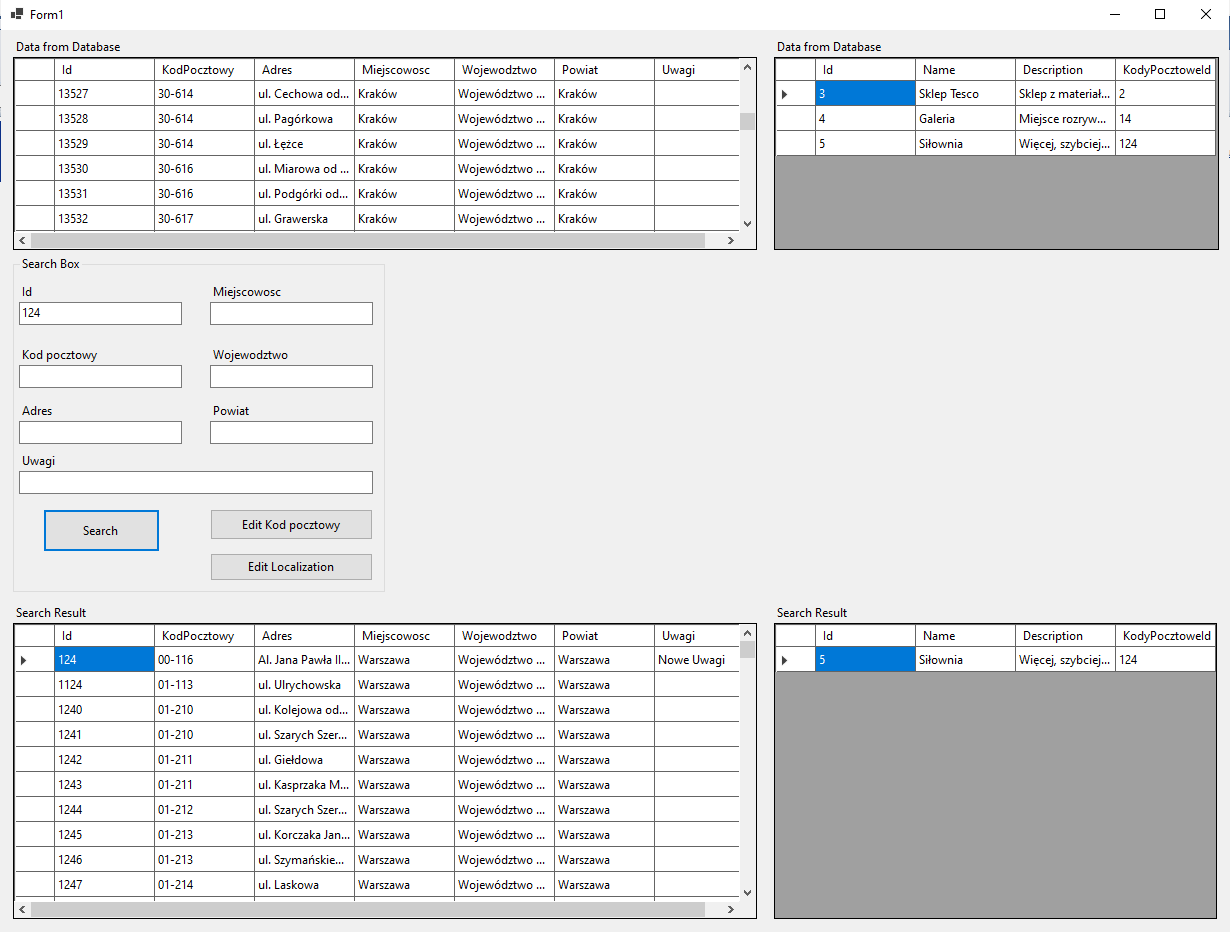
Kliknięcie przycisku akcji w kontrolce (kasowanie, edycja dla kodu pocztowego) powinno wyświetlić komunikat o rezultacie operacji (OK lub Error wraz z podaniem przyczyny), zamknąć kontrolkę (usunąć ją z ekranu) oraz odświeżyć dane w głównym oknie programu. W przypadku dodawania, edycji i kasowania lokalizacji przypisanych do kodu (zagnieżdżony DataGridView na kontrolce), powinno to się odbywać bezpośrednio w kontrolce.

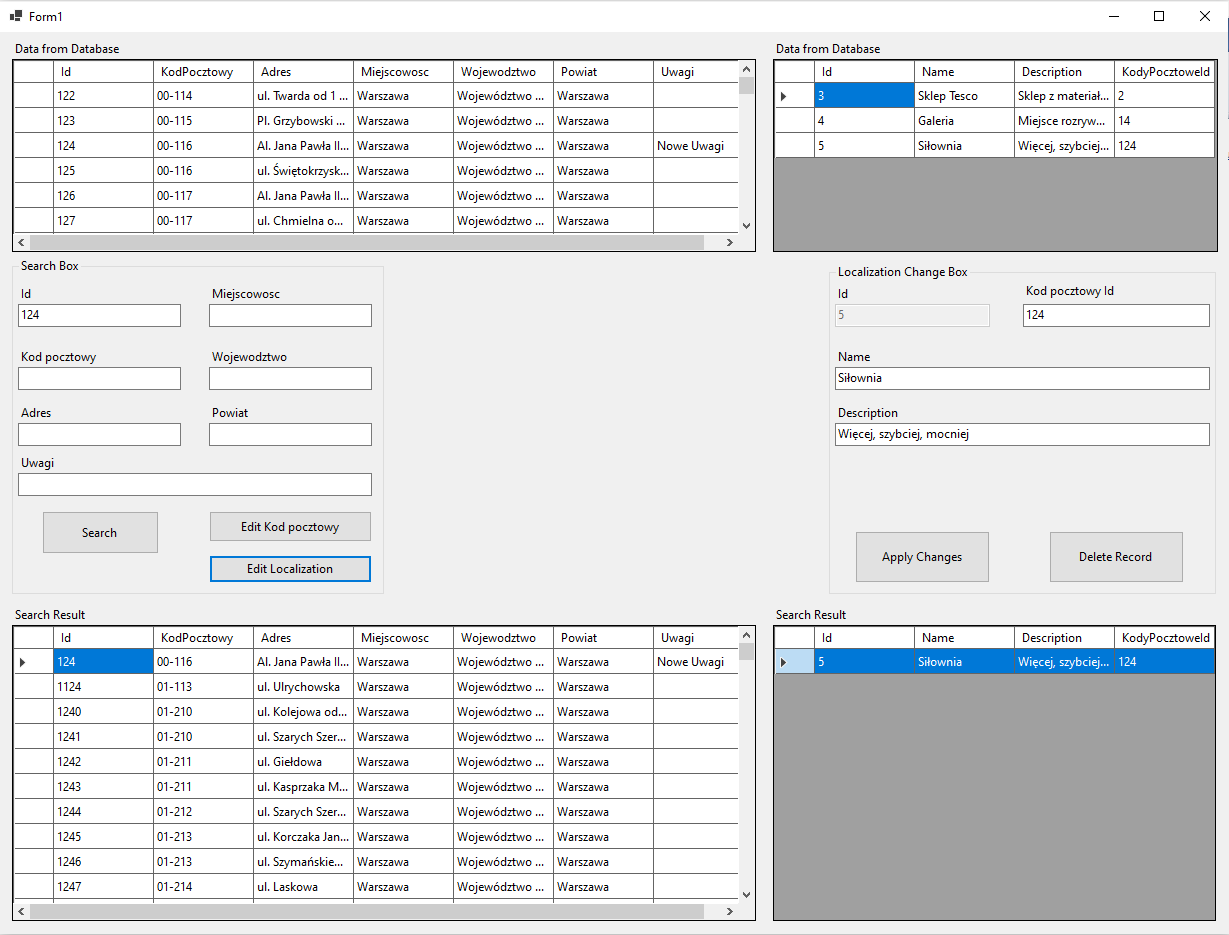
Zadbać o komunikację pomiędzy kontrolką a głównym oknem aplikacji, tak by „wiedziało ono” o tym, że wprowadzono jakiekolwiek zmiany i że lista w głównym oknie programu powinna zostać odświeżona – w celu tym wykorzystać zdarzenia, po utworzeniu kontrolki należy przypisać w formularzu głównym zdarzenie zdefiniowane w tejże kontrolce.

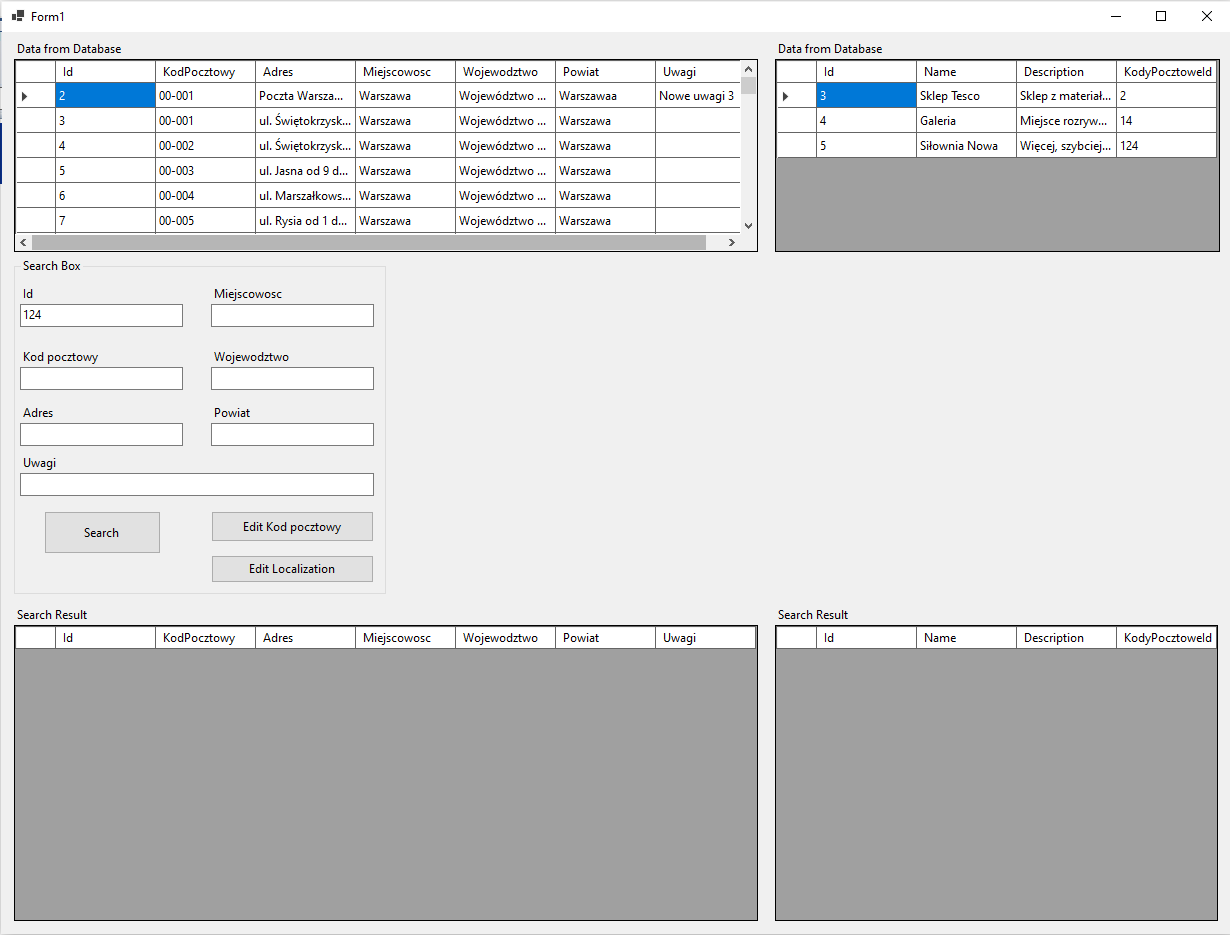
Przeładowanie listy na głównym oknie aplikacji ma się odbywać tylko i wyłącznie jeżeli w trakcie pracy użytkownik wprowadzi jakiekolwiek zmiany – kontrolka zarejestrowała fakt edycji i zakomunikowała to w stosownym zdarzeniu. Oczywiście wszystkie wprowadzane zmiany mają być wykonywane na bazie danych.

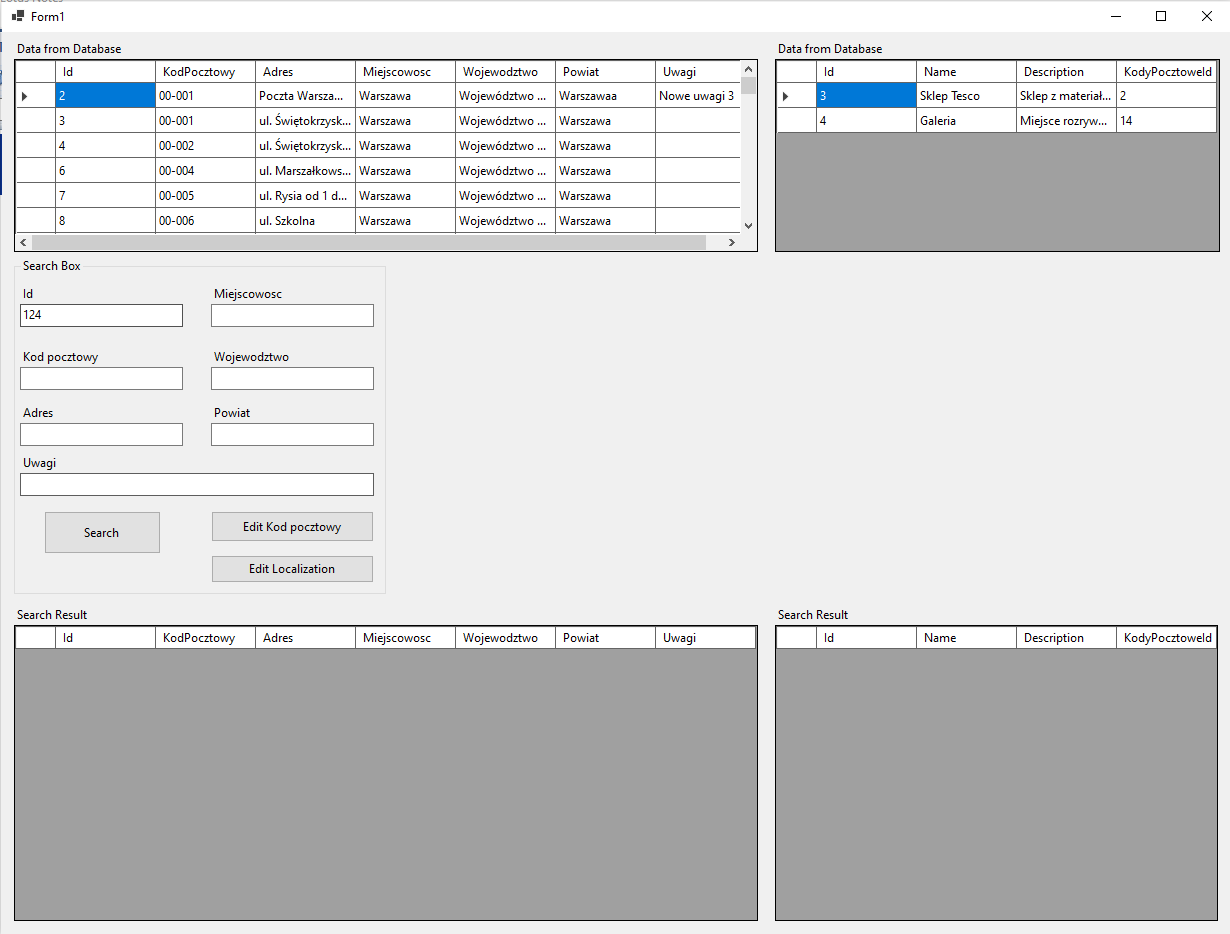


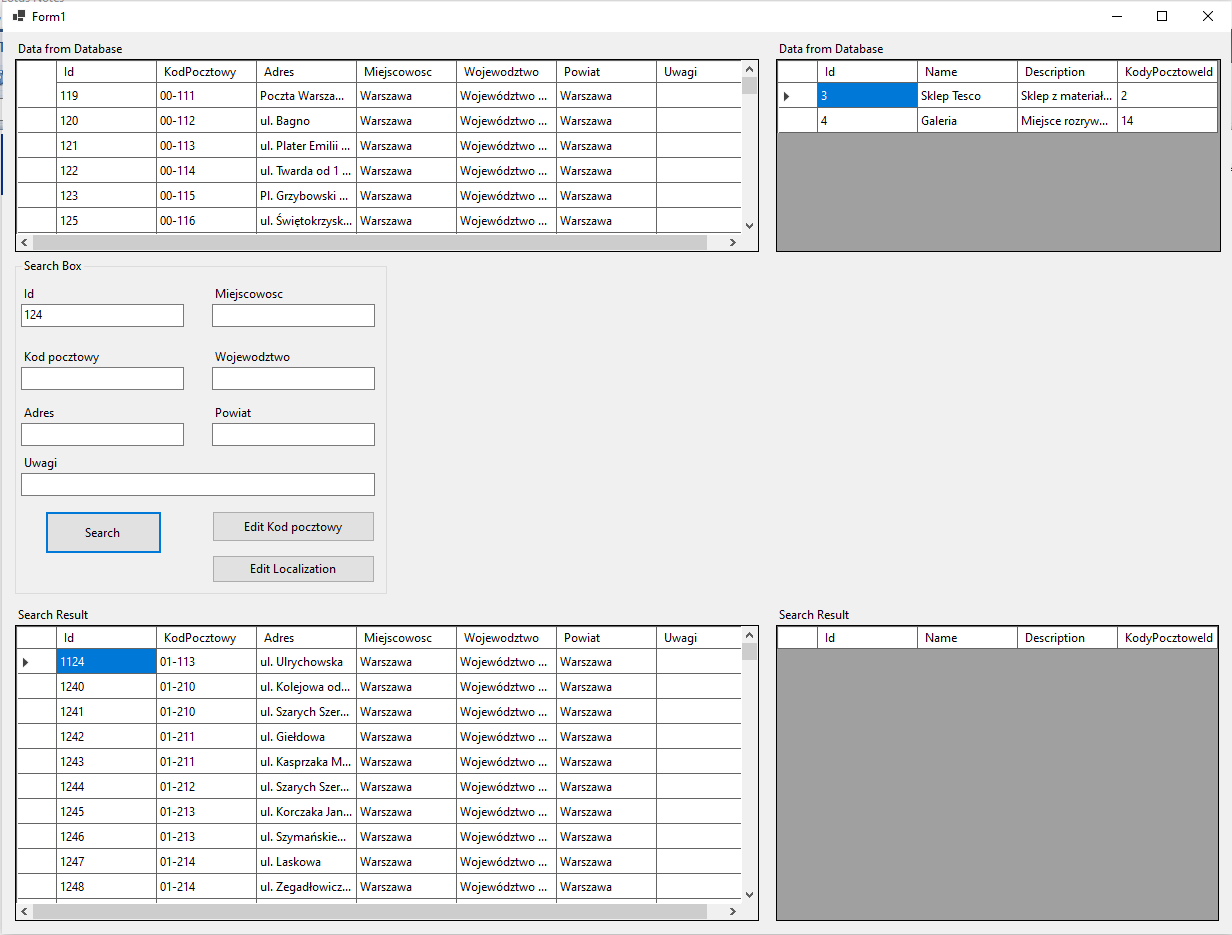












# Wnioski

# Bibliografia

1. Źródła pomocnicze
   1. Guide to ADO.NET and working with databases in .NET 6 (oryg. Руководство по ADO.NET и работе с базами данных в .NET 6) [<https://metanit.com/sharp/adonetcore/>], dostęp: 16.03.2023
   2. Entity Framework Core Series [<https://code-maze.com/entity-framework-core-series/>], dostęp: 16.03.2023
2. Napotkane problem i ich rozwiązania
   1. Tytuł [link], dostęp: dd.mm.rrrr

# Spis ilustracji

# Spis snippetów

**Nie można odnaleźć pozycji dla spisu ilustracji.**