|  |  |  |  |
| --- | --- | --- | --- |
| **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:**  01/07/2023 |
| **Ocena:** | | **Podpis prowadzącego:** |

# Spis treści

[1. Spis treści 1](#_Toc139121579)

[2. Cel ćwiczenia 2](#_Toc139121580)

[3. Wymagania znajomości zagadnień 2](#_Toc139121581)

[4. Literatura, materiały dydaktyczne 2](#_Toc139121582)

[5. Wiadomości teoretyczne 3](#_Toc139121583)

[6. Przebieg ćwiczenia praktycznego 3](#_Toc139121584)

[7. Opracowanie sprawozdania 4](#_Toc139121585)

[8. Wnioski 16](#_Toc139121586)

[9. Bibliografia 16](#_Toc139121587)

[10. Spis ilustracji 16](#_Toc139121588)

[11. Spis snippetów 16](#_Toc139121589)

# 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

–

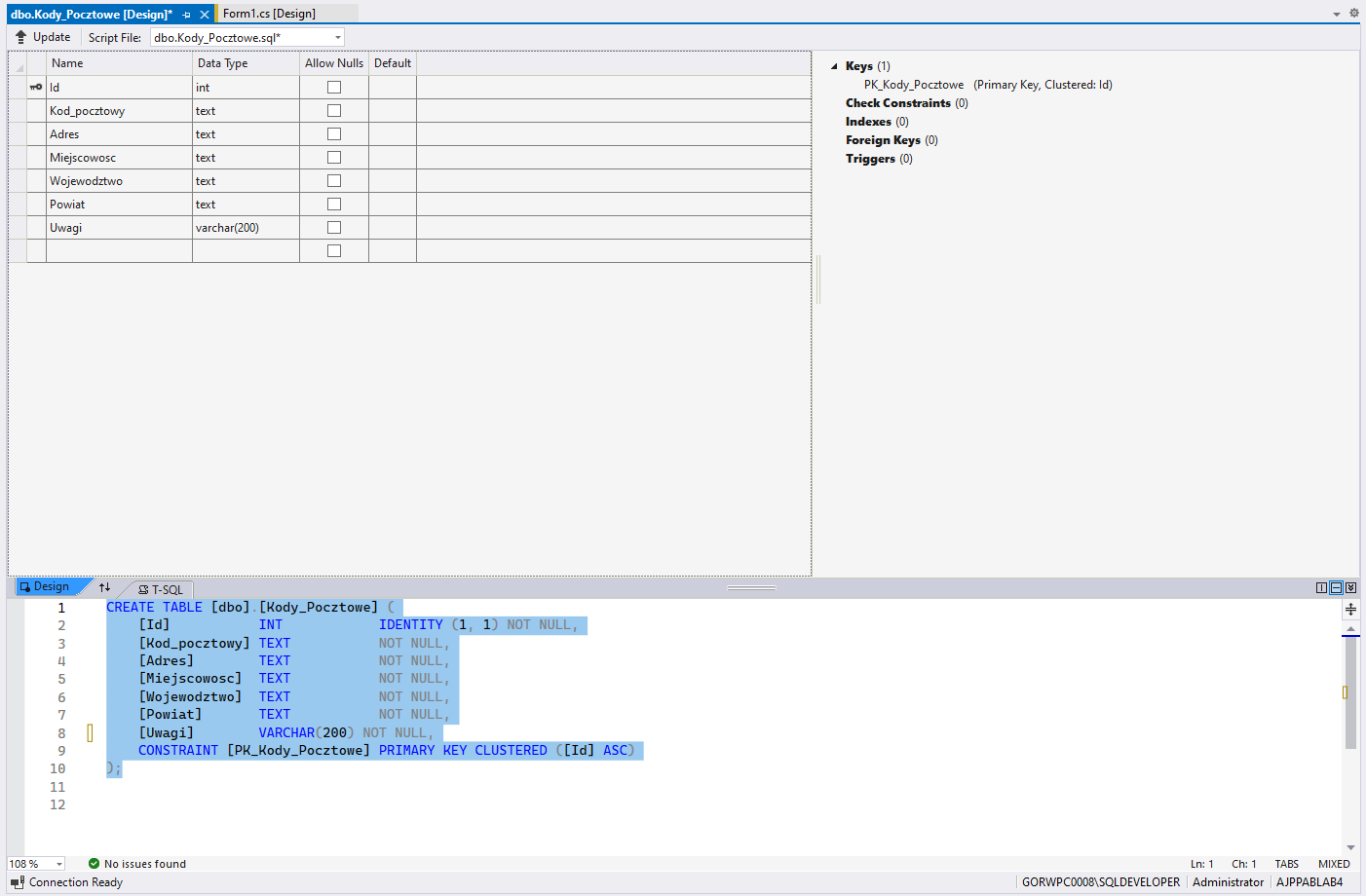
# Przebieg ćwiczenia praktycznego

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.

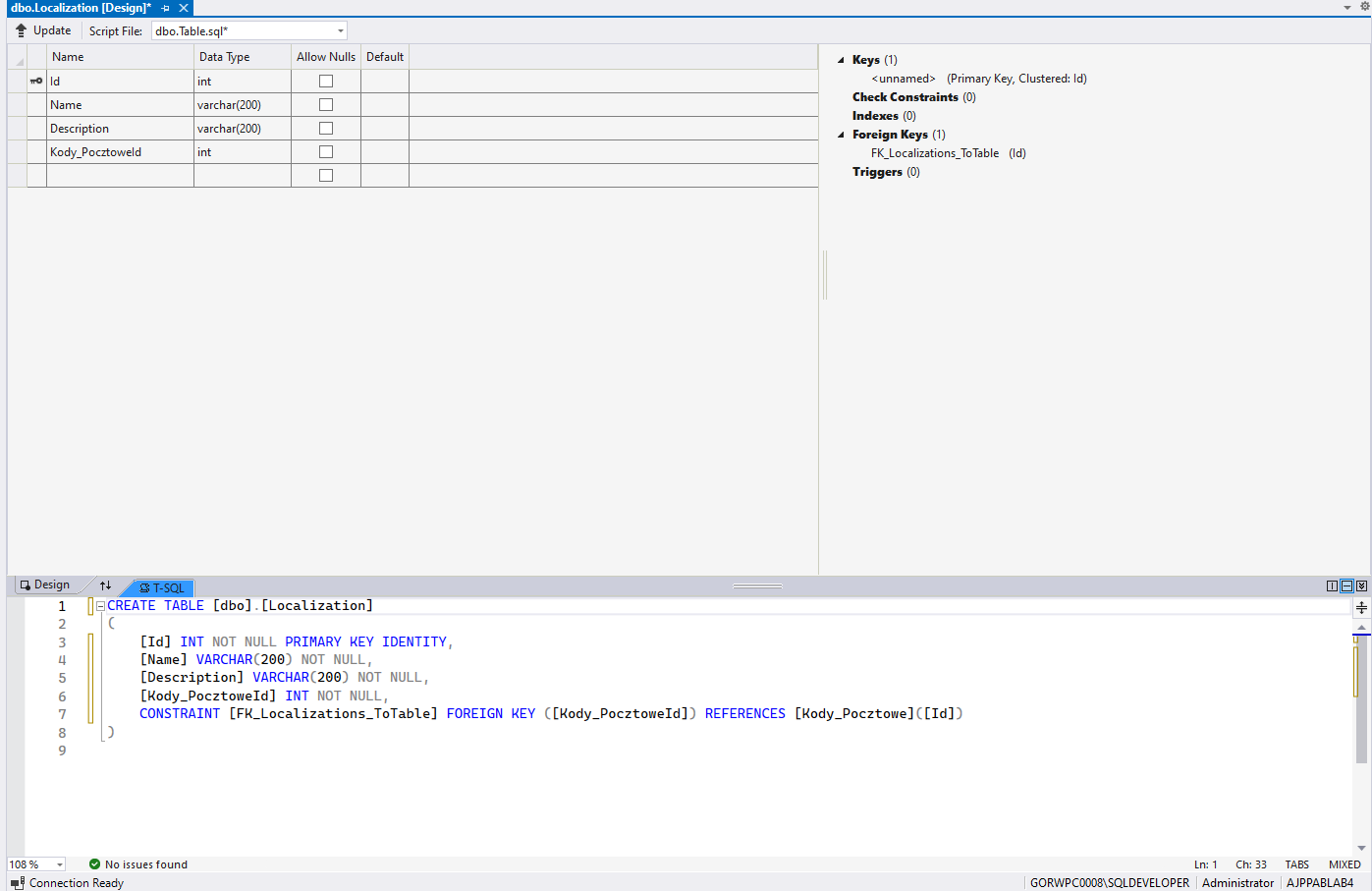
# Opracowanie sprawozdania

Wykorzystując tabele wraz z danymi z poprzedniego laboratorium zmodyfikowane zostały tabele o dwa dodatkowe 2 pola: **Id**, **Uwagi**. Id jest kluczem głównym w tabeli z opcją autoincreent a pole uwag polem Varchar(200). Schemat tej tabeli jest pokazany na Rysunek 1



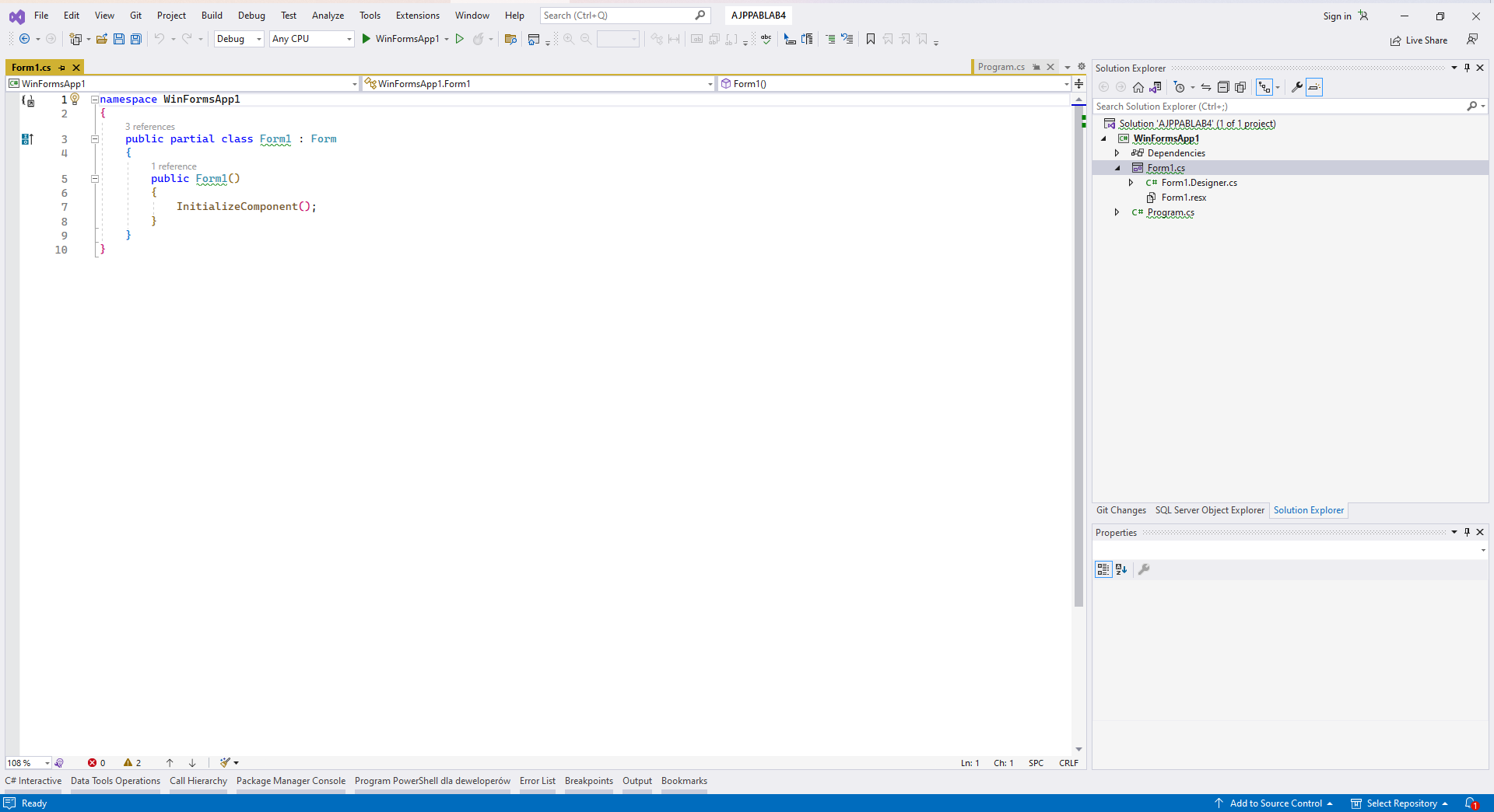
Rysunek . Schemat tabeli Kody\_Pocztowe

Dodatkowo została tabela o nazwie Localization, tabela ta zawiera następujące pola: Id, Name, Description oraz Kody\_PocztoweId (klucz obcy). Schemat tej tabeli jest pokazany na



Rysunek . Schemat tabeli Localization

Utworzony został projekt Window Forms .NET 7 i do niego zostały dołączone pakiety z Entity Framework’iem. Wygląd projektu oraz jego struktura pokazane na Rysunek 3 , a skrypt, który umożliwia zainstalowanie pakietów Nu-Get pokazany na Snippet 1.



Rysunek . Struktura projektu

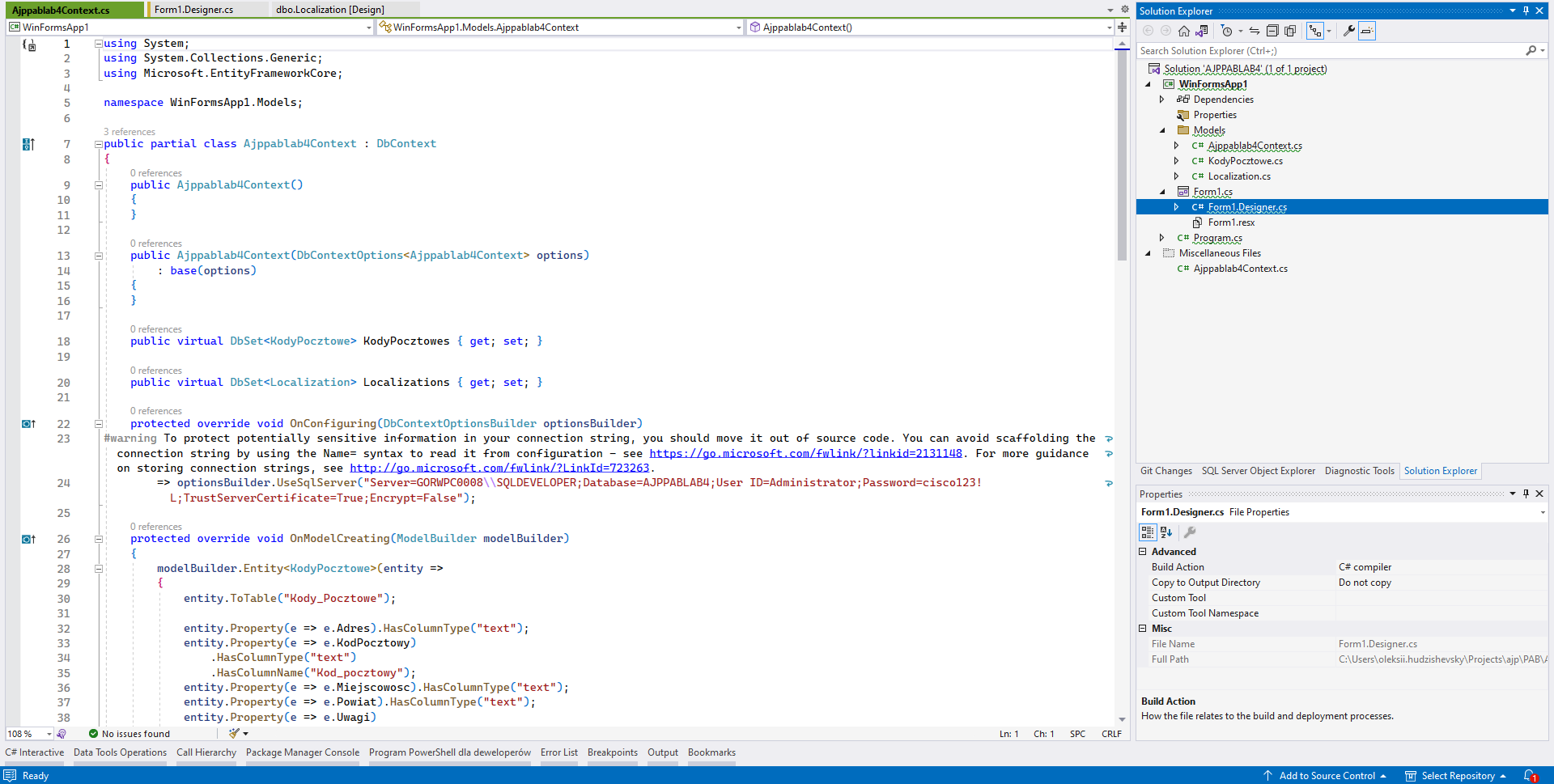
Snippet . Skrypt do zainstalowania EF

|  |
| --- |
| 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 |

Następnie za pomocą polecenia Scaffold utworzony został kontekst bazy danych wraz z tabelami w postaci klas. Skrypt wykonujący generację tego kodu zaprezentowany jest na Snippet 2, a sam wygenerowany kod wraz z plikami pokazany na Rysunek 4.

Snippet . Polecenie Scaffold-DbContext

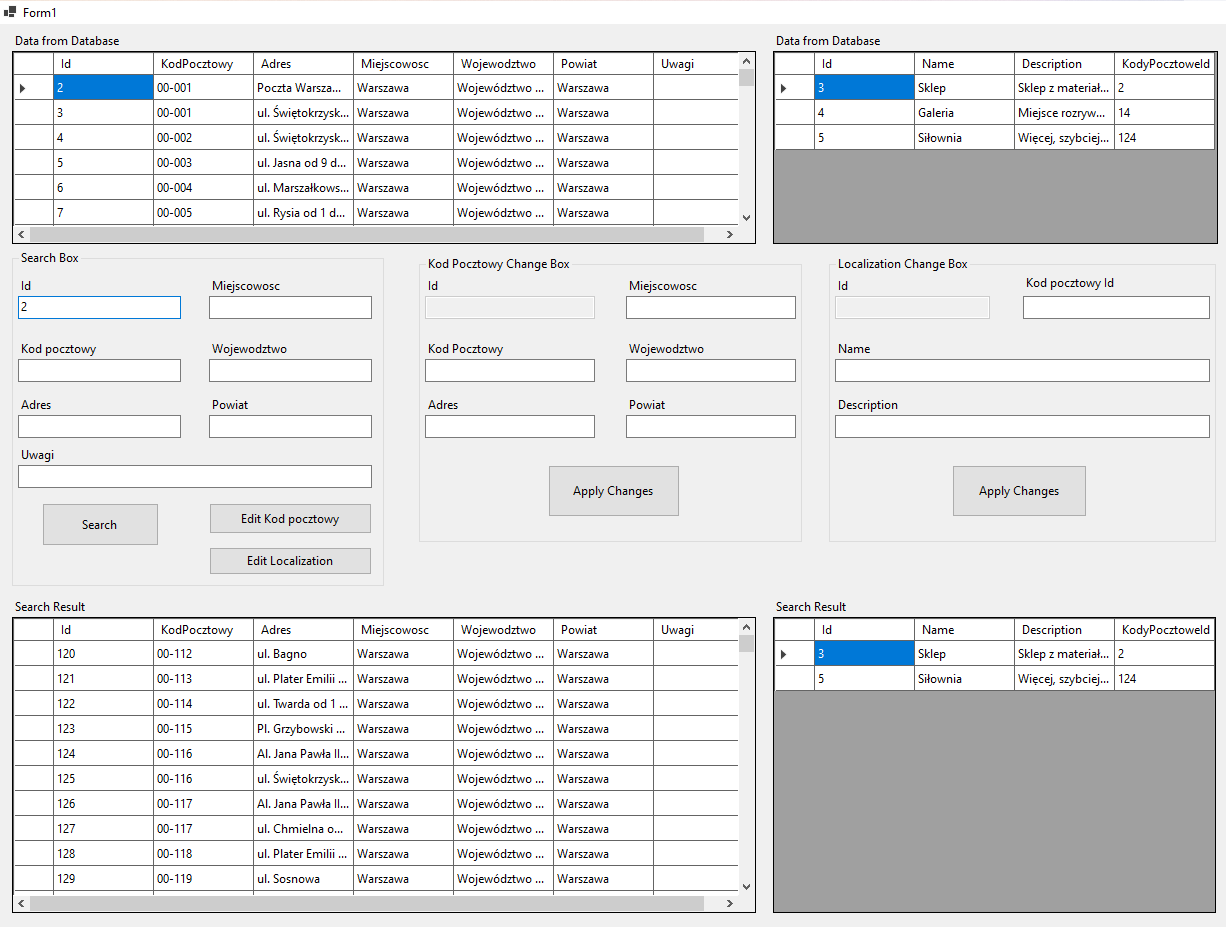
|  |
| --- |
| Scaffold-DbContext "Server=GORWPC0008\SQLDEVELOPER;Database=AJPPABLAB4;User ID=Administrator;Password=cisco123!L;TrustServerCertificate=True;Encrypt=False" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models |



Rysunek . Kontekst bazy danych

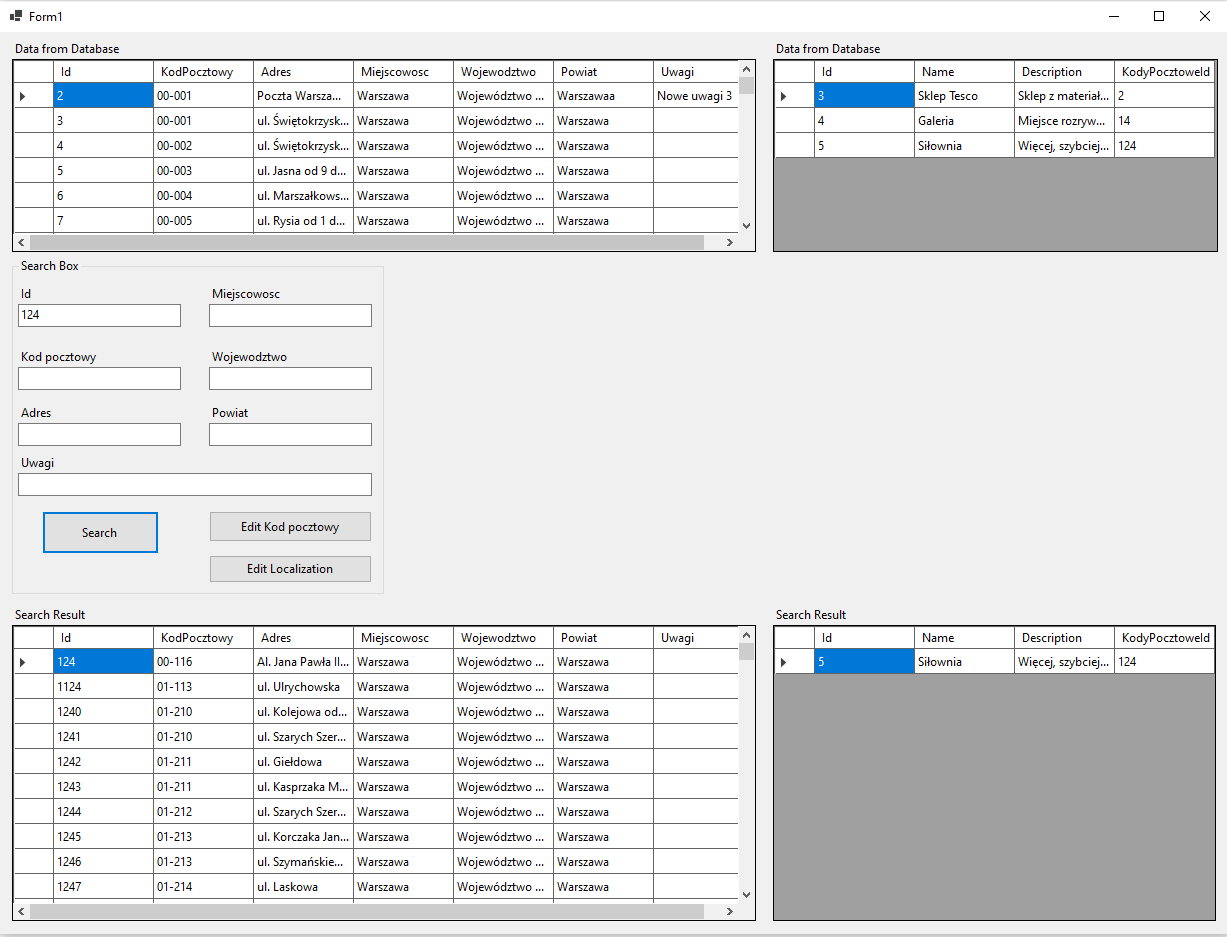
Następnie w pliku Form1 został utworzony widok z kontrolkami do zrealizowania laboratorium. Widok aplikacji jest pokazany Rysunek 5. Widok programu zawiera:

* DataGridView do wyświetlania tabeli Kody\_Pocztowe z bazy danych,
* DataGridView do wyświetlania tabeli Localization z bazy danych,
* DataGridView, który wyświetla wynik wyszukiwania z tabeli Kody\_Pocztowe
* DataGridView, który wyświetla powiązane dane z wyświetlanymi na innym DataGridView (z danymi Kody\_Pocztowe)
* SearchBox zawiera TexBox’y z kryteriami do wyszukiwania konkretnych informacji w tabeli Kody\_Pocztowe,
* Kod\_ Pocztowy Change Box zawiera formularz, w którym możliwa jest modyfikacja wybranego rekordu z tabeli Kody\_Pocztowe lub utworzenie nowego rekordu. Jest ukryty, dopóki nie zostanie aktywowany przyciskiem „Edit Kod\_Pocztowy”,
* Localization Change Box zawiera formularz, w którym możliwa jest modyfikacja wybranego rekordu z tabeli Localization. Jest ukryty, dopóki nie zostanie aktywowany przyciskiem „Edit Localization,



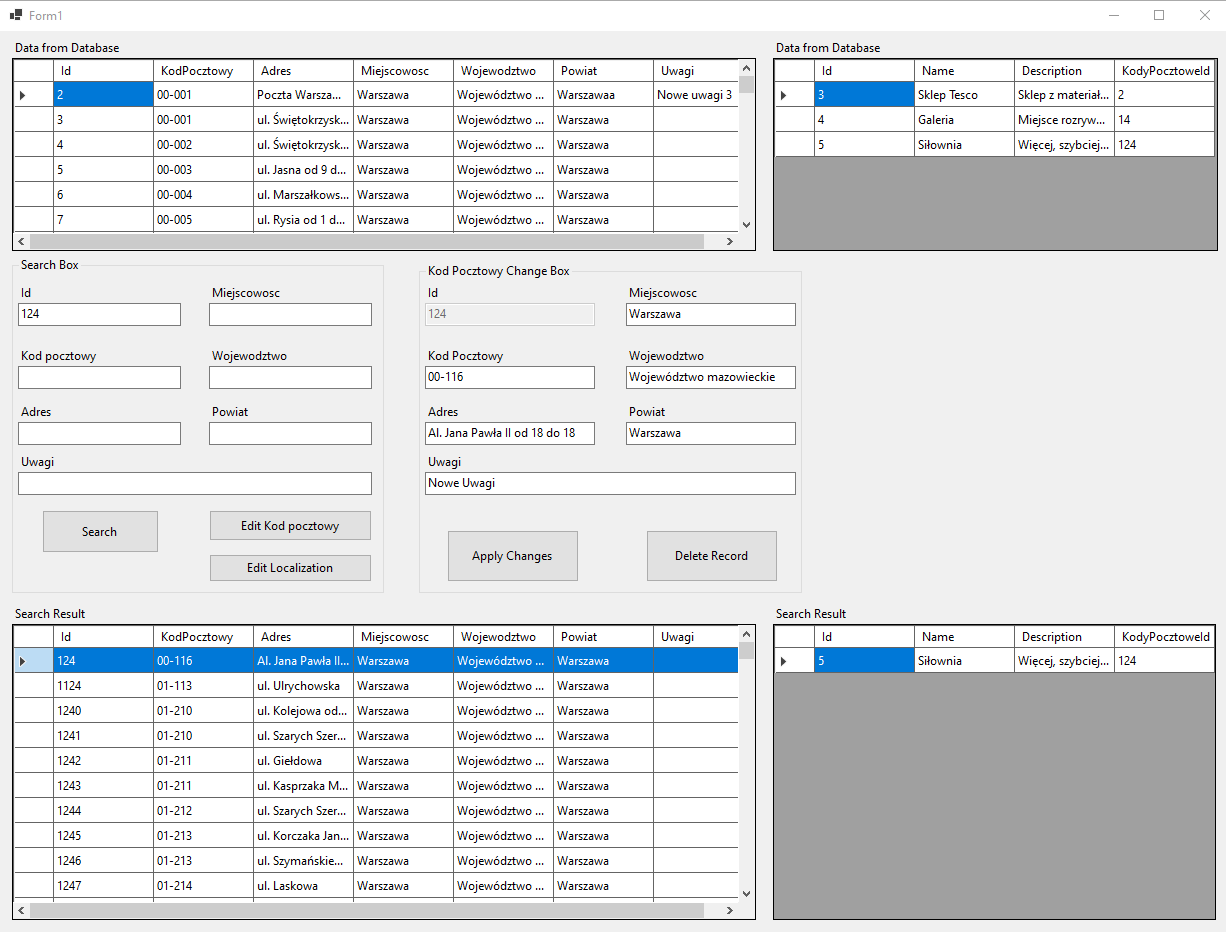
Rysunek . Widok aplikacji

Po wpisaniu wyszukiwanego kryteria i kliknięciu przyciska „Search” w dolnej części zostaną wyświetlone Kody\_Pocztowe i w drugim dolny DataGridView zostaną wyświetlone powiązane z nim lokalizacje. Przykładowe działanie pokazane jest na Rysunek 6. Kod realizujący tą funkcjonalność jest pokazany na Snippet 3.



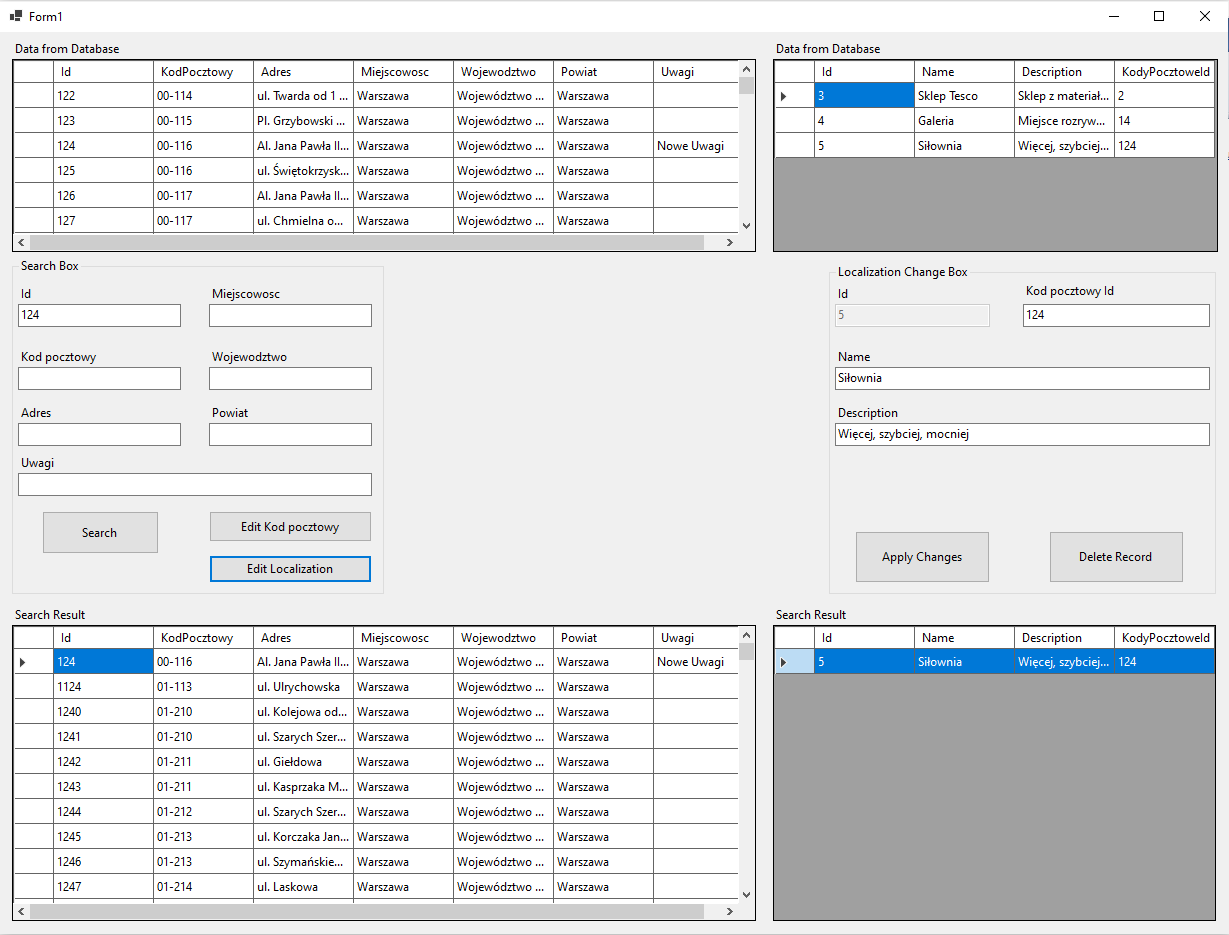
Rysunek . Wyświetlanie powiązanych danych

Po zaznaczeniu wiersza z kodem pocztowym i kliknięciu przyciska „Edit Kod Pocztowy” zostanie wyświetlony dodatkowy formularz z wczytanymi danymi z zaznaczonego wiersza, jeśli żaden wiersz nie był zaznaczony – zostanie wyświetlony pusty formularz, w którym można dodać nowy rekord. Przykładowe działanie pokazane jest na Rysunek 7. Po kliknięciu przyciska „Delete Record” zaznaczony wiersz zostanie usunięty. Kod realizujący te funkcjonalności pokazany jest na Snippet 4 i Snippet 5 i Snippet 6.



Rysunek . Modyfikacja kodu pocztowego

Po zaznaczeniu wiersza z lokalizacją i kliknięciu przyciska „Edit Localization” zostanie wyświetlony dodatkowy formularz z wczytanymi danymi z zaznaczonego wiersza, jeśli żaden wiersz nie był zaznaczony – zostanie wyświetlony pusty formularz, w którym można dodać nowy rekord. Przykładowe działanie pokazane jest na Rysunek 8. Po kliknięciu przyciska „Delete Record” zaznaczony wiersz zostanie usunięty. Kod realizujący te funkcjonalności pokazany jest na Snippet 7 i Snippet 8 i Snippet 9.



Rysunek . Modyfikacja lokalizacji

Snippet . Search Button

|  |
| --- |
| private void searchButton\_Click(object sender, EventArgs e)  {  try  {  string? id = idTextBox.Text.Length > 0 ? idTextBox.Text : null;  string? miejscowosc = miejscowoscTextBox.Text.Length > 0 ? miejscowoscTextBox.Text : null;  string? powiat = powiatTextBox.Text.Length > 0 ? powiatTextBox.Text : null;  string? adres = adresTextBox.Text.Length > 0 ? adresTextBox.Text : null;  string? kodPocztowy = kodPocztowyTextBox.Text.Length > 0 ? kodPocztowyTextBox.Text : null;  string? wojewodztwo = wojewodztwoTextBox.Text.Length > 0 ? wojewodztwoTextBox.Text : null;  string? uwagi = uwagiTextBox.Text.Length > 0 ? uwagiTextBox.Text : null;  List<KodyPocztowe> result = new();  List<Localization> localizationsResult = new();  Stopwatch stopwatch = new();  stopwatch.Start();  using (Ajppablab4Context db = new())  {  result = db.KodyPocztowes.ToList();  if (id != null)  {  result = result.Where(kod => kod.Id.ToString().ToLower().Contains(id.ToLower())).ToList();  }  if (miejscowosc != null)  {  result = result.Where(kod => kod.Miejscowosc.ToLower().Contains(miejscowosc.ToLower())).ToList();  }  if (powiat != null)  {  result = result.Where(kod => kod.Powiat.ToLower().Contains(powiat.ToLower())).ToList();  }  if (adres != null)  {  result = result.Where(kod => kod.Adres.ToLower().Contains(adres.ToLower())).ToList();  }  if (kodPocztowy != null)  {  result = result.Where(kod => kod.KodPocztowy.ToLower().Contains(kodPocztowy.ToLower())).ToList();  }  if (wojewodztwo != null)  {  result = result.Where(kod => kod.Wojewodztwo.ToLower().Contains(wojewodztwo.ToLower())).ToList();  }  if (wojewodztwo != null)  {  result = result.Where(kod => kod.Uwagi.ToLower().Contains(uwagi.ToLower())).ToList();  }  if (result.Count < 0)  {  throw new Exception("Nothing Found");  }  if (result.Count > 0)  {  foreach (var item in result)  {  localizationsResult.AddRange(db.Localizations.Where(l => l.KodyPocztoweId == item.Id).ToList());  }  }  }  stopwatch.Stop();  kodyPocztoweSearchedBindingSource.DataSource = result;  localizationSearchedBindingSource.DataSource = localizationsResult;  MessageBox.Show($"Search Result: {result.Count} items \nIn time: {stopwatch.Elapsed}");  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  }  } |

Snippet . Edit Kod\_Pocztowy Button

|  |
| --- |
| private void editKodPocztowyButton\_Click(object sender, EventArgs e)  {  try  {  kodPocztowyChangeGroupBox.Visible = true;  var selectedRows = kodPocztowySearchedGridView.SelectedRows;  if (selectedRows.Count > 0)  {  var selectedRow = selectedRows[0];  changeKodPocztowyIdTextBox.Text = selectedRow.Cells[0].Value.ToString();  changeKodPocztowyTextBox.Text = selectedRow.Cells[1].Value.ToString();  changeKodPocztowyAdresTextBox.Text = selectedRow.Cells[2].Value.ToString();  changeKodPocztowyMiejscowoscTextBox.Text = selectedRow.Cells[3].Value.ToString();  changeKodPocztowyWojewodztwoTextBox.Text = selectedRow.Cells[4].Value.ToString();  changeKodPocztowyPowiatTextBox.Text = selectedRow.Cells[5].Value.ToString();  changeKodPocztowyUwagiTextBox.Text = selectedRow.Cells[6].Value.ToString();  }  else  {  MessageBox.Show($"You have not selected any row to modify, and now a form to create a new object is displayed");  }  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  kodPocztowyChangeGroupBox.Visible = false;  }  } |

Snippet . Change Kod\_Pocztowy Button

|  |
| --- |
| private void changesKodPocztowyButton\_Click(object sender, EventArgs e)  {  try  {  if (changeKodPocztowyIdTextBox.Text == "")  {  using (Ajppablab4Context db = new())  {  KodyPocztowe newKodPoczotowy = new()  {  KodPocztowy = changeKodPocztowyTextBox.Text,  Miejscowosc = changeKodPocztowyMiejscowoscTextBox.Text,  Wojewodztwo = changeKodPocztowyWojewodztwoTextBox.Text,  Adres = changeKodPocztowyAdresTextBox.Text,  Powiat = changeKodPocztowyPowiatTextBox.Text,  Uwagi = changeKodPocztowyUwagiTextBox.Text,  };  db.KodyPocztowes.Add(newKodPoczotowy);  db.SaveChanges();  }  }  else  {  int? id = null;  if (Int32.TryParse(changeKodPocztowyIdTextBox.Text, out \_)) id = Int32.Parse(changeKodPocztowyIdTextBox.Text);  using (Ajppablab4Context db = new())  {  var updatedKodPocztowy = db.KodyPocztowes.Where(kp => kp.Id == id).FirstOrDefault();  updatedKodPocztowy.KodPocztowy = changeKodPocztowyTextBox.Text;  updatedKodPocztowy.Miejscowosc = changeKodPocztowyMiejscowoscTextBox.Text;  updatedKodPocztowy.Wojewodztwo = changeKodPocztowyWojewodztwoTextBox.Text;  updatedKodPocztowy.Adres = changeKodPocztowyAdresTextBox.Text;  updatedKodPocztowy.Powiat = changeKodPocztowyPowiatTextBox.Text;  updatedKodPocztowy.Uwagi = changeKodPocztowyUwagiTextBox.Text;  db.SaveChanges();  }  }  clearKodyResultDataGridView();  clearLocalizationsResultDataGridView();  fillKodyDataGridView();  fillLocalizationsDataGridView();  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  }  finally  {  changeKodPocztowyTextBox.Text = "";  changeKodPocztowyMiejscowoscTextBox.Text = "";  changeKodPocztowyWojewodztwoTextBox.Text = "";  changeKodPocztowyAdresTextBox.Text = "";  changeKodPocztowyPowiatTextBox.Text = "";  changeKodPocztowyUwagiTextBox.Text = "";  kodPocztowyChangeGroupBox.Visible = false;  }  } |

Snippet . Delet Kod\_Pocztowy Button

|  |
| --- |
| private void deleteKodPocztowyButton\_Click(object sender, EventArgs e)  {  try  {  int? id = null;  if (Int32.TryParse(changeKodPocztowyIdTextBox.Text, out \_)) id = Int32.Parse(changeKodPocztowyIdTextBox.Text);  using (Ajppablab4Context db = new())  {  var deletedKodPocztowy = db.KodyPocztowes.Where(kp => kp.Id == id).FirstOrDefault();  db.KodyPocztowes.Remove(deletedKodPocztowy);  db.SaveChanges();  }  clearKodyResultDataGridView();  clearLocalizationsResultDataGridView();  fillKodyDataGridView();  fillLocalizationsDataGridView();  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  }  finally  {  changeKodPocztowyTextBox.Text = "";  changeKodPocztowyMiejscowoscTextBox.Text = "";  changeKodPocztowyWojewodztwoTextBox.Text = "";  changeKodPocztowyAdresTextBox.Text = "";  changeKodPocztowyPowiatTextBox.Text = "";  changeKodPocztowyUwagiTextBox.Text = "";  kodPocztowyChangeGroupBox.Visible = false;  }  } |

Snippet . Edit Localization Button

|  |
| --- |
| private void editLocalizationButton\_Click(object sender, EventArgs e)  {  try  {  localizationChangeGroupBox.Visible = true;  var selectedRows = localizationsSearchedGridView.SelectedRows;  if (selectedRows.Count > 0)  {  var selectedRow = selectedRows[0];  changeLocalizationIdTextBox.Text = selectedRow.Cells[0].Value.ToString();  changeLocalizationKodPocztowyIdTextBox.Text = selectedRow.Cells[3].Value.ToString();  changeLocalizationNameTextBox.Text = selectedRow.Cells[1].Value.ToString();  changeLocalizationDescriptionTextBox.Text = selectedRow.Cells[2].Value.ToString();  }  else  {  MessageBox.Show($"You have not selected any row to modify, and now a form to create a new object is displayed");  }  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  localizationChangeGroupBox.Visible = true;  }  } |

Snippet . Changes Localization Button

|  |
| --- |
| private void changesLocalizationButton\_Click(object sender, EventArgs e)  {  try  {  if (changeLocalizationIdTextBox.Text == "")  {  using (Ajppablab4Context db = new())  {  int id;  if (Int32.TryParse(changeLocalizationKodPocztowyIdTextBox.Text, out \_))  {  id = Int32.Parse(changeLocalizationKodPocztowyIdTextBox.Text);  var foreingKodPocztowy = db.KodyPocztowes.Where(kp => kp.Id == id).FirstOrDefault();  if (foreingKodPocztowy == null)  {  MessageBox.Show($"You are trying to bind a new object to a non-existent 'Kode Pocztowy'");  }  else  {  Localization newLocalization = new()  {  KodyPocztoweId = id,  Name = changeLocalizationNameTextBox.Text,  Description = changeLocalizationDescriptionTextBox.Text  };  db.Localizations.Add(newLocalization);  db.SaveChanges();  }  }  else  {  MessageBox.Show($"'Kod Pocztowy Id' must be Integer");  }  }  }  else  {  int id = Int32.Parse(changeLocalizationIdTextBox.Text);  using (Ajppablab4Context db = new())  {  var updatedLocalization = db.Localizations.Where(l => l.Id == id).FirstOrDefault();  int kodPocztowyId;  if (Int32.TryParse(changeLocalizationIdTextBox.Text, out \_))  {  kodPocztowyId = Int32.Parse(changeLocalizationKodPocztowyIdTextBox.Text);  var foreingKodPocztowy = db.KodyPocztowes.Where(kp => kp.Id == id).FirstOrDefault();  if (foreingKodPocztowy == null)  {  MessageBox.Show($"You are trying to bind a new object to a non-existent 'Kode Pocztowy'");  }  else  {  updatedLocalization.KodyPocztoweId = kodPocztowyId;  updatedLocalization.Name = changeLocalizationNameTextBox.Text;  updatedLocalization.Description = changeLocalizationDescriptionTextBox.Text;  db.SaveChanges();  }  }  else  {  MessageBox.Show($"'Kod Pocztowy Id' must be Integer");  }  }  }  clearKodyResultDataGridView();  clearLocalizationsResultDataGridView();  fillKodyDataGridView();  fillLocalizationsDataGridView();  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  }  finally  {  changeLocalizationIdTextBox.Text = "";  changeLocalizationDescriptionTextBox.Text = "";  changeLocalizationNameTextBox.Text = "";  changeLocalizationKodPocztowyIdTextBox.Text = "";  localizationChangeGroupBox.Visible = false;  }  } |

Snippet . Delete Localization Button

|  |
| --- |
| private void deleteLocalizationButton\_Click(object sender, EventArgs e)  {  try  {  using (Ajppablab4Context db = new())  {  int? id = null;  if (Int32.TryParse(changeLocalizationIdTextBox.Text, out \_)) id = Int32.Parse(changeLocalizationIdTextBox.Text);  var deletedLocalization = db.Localizations.Where(kp => kp.Id == id).FirstOrDefault();  db.Localizations.Remove(deletedLocalization);  db.SaveChanges();  }  clearKodyResultDataGridView();  clearLocalizationsResultDataGridView();  fillKodyDataGridView();  fillLocalizationsDataGridView();  }  catch (Exception ex)  {  MessageBox.Show($"Occured error {ex.Message}");  }  finally  {  changeLocalizationIdTextBox.Text = "";  changeLocalizationDescriptionTextBox.Text = "";  changeLocalizationNameTextBox.Text = "";  changeLocalizationKodPocztowyIdTextBox.Text = "";  localizationChangeGroupBox.Visible = false;  }  } |

# Wnioski

W trakcie wykonania laboratorium udało się jeszcze bardziej poszerzyć wiedzę zarówno jak praktyczną i teoretyczną w zakresie narzędzi ORM i aplikacji typu Windows Forms, a także przećwiczyć wymianę informacji pomiędzy kontrolkami a bazą danych.

# Bibliografia

1. Źródła pomocnicze
   1. Dokumentacja Microsoft [<https://learn.microsoft.com/>], dostęp: 25.06.2023
   2. Kolumny w DataGrid [<https://wpf-tutorial.com/pl/89/kontrolka-datagrid/kolumny-w-datagrid/>], dostęp: 25.06.2023
   3. 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: 25.06.2023
   4. Entity Framework Core Series [<https://code-maze.com/entity-framework-core-series/>], dostęp: 25.06.2023

# Spis ilustracji

[Rysunek 1. Schemat tabeli Kody\_Pocztowe 5](#_Toc139121474)

[Rysunek 2. Schemat tabeli Localization 5](#_Toc139121475)

[Rysunek 3. Struktura projektu 6](#_Toc139121476)

[Rysunek 4. Kontekst bazy danych 7](#_Toc139121477)

[Rysunek 5. Widok aplikacji 8](#_Toc139121478)

[Rysunek 6. Wyświetlanie powiązanych danych 8](#_Toc139121479)

[Rysunek 7. Modyfikacja kodu pocztowego 9](#_Toc139121480)

[Rysunek 8. Modyfikacja lokalizacji 10](#_Toc139121481)

# Spis snippetów

[Snippet 1. Skrypt do zainstalowania EF 6](#_Toc139121482)

[Snippet 2. Polecenie Scaffold-DbContext 6](#_Toc139121483)

[Snippet 3. Search Button 10](#_Toc139121484)

[Snippet 4. Edit Kod\_Pocztowy Button 11](#_Toc139121485)

[Snippet 5. Change Kod\_Pocztowy Button 12](#_Toc139121486)

[Snippet 6. Delet Kod\_Pocztowy Button 13](#_Toc139121487)

[Snippet 7. Edit Localization Button 13](#_Toc139121488)

[Snippet 8. Changes Localization Button 14](#_Toc139121489)

[Snippet 9. Delete Localization Button 15](#_Toc139121490)