

Bazy danych II.
Projekt 6: Przetwarzanie własnych typów danych CLR
UDT.

System rekrutacyjny umożliwiający zarządzanie
kandydatami, rozmowami kwalifikacyjnymi, ocenami i
zadaniami technicznymi.

Yuliya Zviarko
(WFiIS AGH)

07.06.2025

Spis treści

| | | |
|---|---|----|
| 1 | Opis problemu | 3 |
| 2 | Udostępnione funkcjonalności | 3 |
| 3 | Typy danych oraz metody udostępnione w ramach API | 9 |
| 4 | Implementacja | 11 |
| 5 | Prezentacja przeprowadzonych testów jednostkowych | 11 |
| 6 | Podsumowanie i wnioski | 12 |
| 7 | Kod źródłowy | 12 |
| 8 | Literatura | 12 |

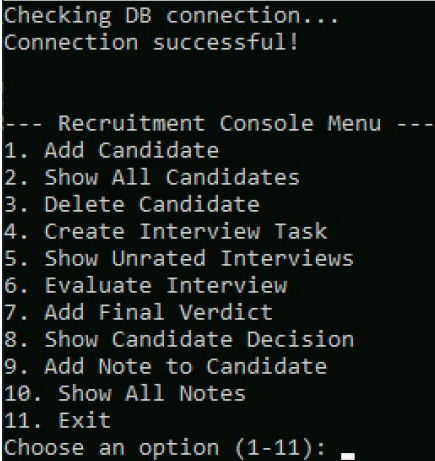
1 Opis problemu

Celem projektu było opracowanie API oraz jego implementacji obsługującej zestaw typów danych UDT w technologii CLR, służącej do obsługi typów złożonych. Całość ma być dostępna z poziomu aplikacji konsolowej, a metody utworzonych typów danych są weryfikowane za pomocą testów jednostkowych.

Stworzone typy danych służą do pracy w obszarze HR, w szczególności do obsługi kandydatów na stanowiska pracy. W skład tych typów wchodzi:

- Candidate,
- TaskItem,
- Interview,
- Evaluation,
- RecruitmentSummary,
- HrNote.

2 Udostępnione funkcjonalności

A screenshot of a console application window with a black background and white text. The text shows the application's startup sequence: 'Checking DB connection...' followed by 'Connection successful!'. Then, a separator '--- Recruitment Console Menu ---' is displayed. Below it is a numbered list of 11 options: 1. Add Candidate, 2. Show All Candidates, 3. Delete Candidate, 4. Create Interview Task, 5. Show Unrated Interviews, 6. Evaluate Interview, 7. Add Final Verdict, 8. Show Candidate Decision, 9. Add Note to Candidate, 10. Show All Notes, and 11. Exit. At the bottom, the prompt 'Choose an option (1-11):' is followed by a white cursor line.

```
Checking DB connection...
Connection successful!

--- Recruitment Console Menu ---
1. Add Candidate
2. Show All Candidates
3. Delete Candidate
4. Create Interview Task
5. Show Unrated Interviews
6. Evaluate Interview
7. Add Final Verdict
8. Show Candidate Decision
9. Add Note to Candidate
10. Show All Notes
11. Exit
Choose an option (1-11): _
```

Rysunek 1: Główny widok aplikacji konsolowej.

Z poziomu aplikacji konsolowej użytkownik uzyskuje dostęp do następujących funkcji: wprowadzania danych do zdefiniowanych struktur, wyszukiwania informacji w tych strukturach, tworzenia odpowiednich raportów, a także informowania o błędnym wykorzystaniu danych.

Kolejne funkcjonalności wybierane są poprzez wpisanie odpowiadającej im cyfry i zatwierdzenie klawiszem Enter. W przypadku wybrania nieistniejącej opcji, aplikacja wyświetli odpowiedni komunikat o błędzie i umożliwi ponowny wybór prawidłowej funkcji.

Dostępne opcje to:

- **1. Add Candidate** – dodanie kandydata do bazy danych. Użytkownik wprowadza imię i nazwisko oraz adres e-mail kandydata. Dane są walidowane – imię i nazwisko mogą zawierać wyłącznie litery i muszą składać się z dwóch wyrazów, a adres e-mail jest sprawdzany pod kątem poprawnego formatu. Operację można przerwać w dowolnym momencie, wpisując q. Po pomyślnym dodaniu wyświetlany jest komunikat potwierdzający, a w przypadku błędów użytkownik otrzymuje stosowną informację.

```
Choose an option (1-11): 1
Enter candidate name (first and last name) or 'q' to cancel: Szarlota Ejczison
Enter candidate email or 'q' to cancel: itscharlibabe@xcx.com
Inserted Candidate.
```

Rysunek 2: Add Candidate opcja.

```
11. EXIT
Choose an option (1-11): 1
Enter candidate name (first and last name) or 'q' to cancel: Imie
Invalid name. Please enter a first and last name using only letters.
```

Rysunek 3: Walidacja imienia i nazwiska.

```
11. EXIT
Choose an option (1-11): 1
Enter candidate name (first and last name) or 'q' to cancel: Imie Nazwisko
Enter candidate email or 'q' to cancel: test@
Error: The email address you entered is invalid.
```

Rysunek 4: Walidacja adresu e-mail.

- **2. Show All Candidates** – wyświetlenie listy wszystkich aktualnie zarejestrowanych kandydatów.

```

Choose an option (1-11): 2

Candidates:
[9] Pawel Sipko - sipko@gmail.com
[10] Mirosław Kolodziej - kolmir@gmail.com
[11] Patrycja Gorka - gorpar@test.com
[12] Adam Laba - laba@onet.pl
[13] Patryk Tarkowski - patryyyk@gmail.com
[14] Krzysztof Nowak - krzysznow@gmail.com
[15] Michał Szuplakow - szupmich@agh.edu.pl
[16] Bartosz Rybak - ryba@gmail.com
[17] Kamil Jagodzinski - kjag@agh.edu.pl
[18] Jakub Wojcik - wojjjjcik@gmail.com
[19] Natalia Tomaszewska - tomnat@agh.edu.pl
[21] Anna Dudowska - dudka@com.pl
[23] Magdalena Szewczyk - szewczyk@test.com
[24] Szarlota Ejczison - itscharlibabe@xcx.com

```

Rysunek 5: Lista kandydatów.

- **3. Delete Candidate** – usunięcie kandydata z bazy danych. Aplikacja wyświetla listę aktualnych kandydatów, a użytkownik podaje ID kandydata, którego chce usunąć. W przypadku podania nieprawidłowego ID pojawia się komunikat o błędzie. Operację można przerwać w dowolnym momencie.

```

Choose an option (1-11): 3

Candidates:
[9] Pawel Sipko - sipko@gmail.com
[10] Mirosław Kolodziej - kolmir@gmail.com
[11] Patrycja Gorka - gorpar@test.com
[12] Adam Laba - laba@onet.pl
[13] Patryk Tarkowski - patryyyk@gmail.com
[14] Krzysztof Nowak - krzysznow@gmail.com
[15] Michał Szuplakow - szupmich@agh.edu.pl
[16] Bartosz Rybak - ryba@gmail.com
[17] Kamil Jagodzinski - kjag@agh.edu.pl
[18] Jakub Wojcik - wojjjjcik@gmail.com
[19] Natalia Tomaszewska - tomnat@agh.edu.pl
[21] Anna Dudowska - dudka@com.pl
[23] Magdalena Szewczyk - szewczyk@test.com
[24] Szarlota Ejczison - itscharlibabe@xcx.com
Enter the Id of the candidate you want to delete or 'q' to cancel: 18
Candidate deleted successfully.

```

Rysunek 6: Usuwanie kandydata.

- **4. Create Interview Task** – tworzenie zadania rekrutacyjnego dla kandydata. Użytkownik wybiera kandydata, przypisuje mu zadanie, określa osobę odpowiedzialną, ustawia datę, tytuł oraz opis zadania. Po pomyślnym zakończeniu operacji wyświetlany jest komunikat oraz ID nowo utworzonego zadania. Wprowadzane dane są walidowane, zwłaszcza data (np. nie można ustawić daty z przeszłości lub nieistniejącej).

```
Choose an option (1-11): 4
Candidates:
[9] Pawel Sipko - sipko@gmail.com
[10] Mirosław Kolodziej - kolmir@gmail.com
[11] Patrycja Gorka - gorpar@test.com
[12] Adam Laba - laba@onet.pl
[13] Patryk Tarkowski - patryyyyk@gmail.com
[14] Krzysztof Nowak - krzysznow@gmail.com
[15] Michał Szuplakow - szupmich@agh.edu.pl
[16] Bartosz Rybak - ryba@gmail.com
[17] Kamil Jagodzinski - kjag@agh.edu.pl
[19] Natalia Tomaszewska - tomat@agh.edu.pl
[21] Anna Dudowska - dudka@com.pl
[23] Magdalena Szewczyk - szewczyk@test.com
[24] Szarlota Ejczison - itscharlibabe@xcx.com
Candidate ID (or 'q' to cancel): 9
Interviewer Name (or 'q' to cancel): Yuliya Zviarko
Interview Date (yyyy-MM-dd) or 'q' to cancel: 2025-06-10
Task Title (or 'q' to cancel): LeetCode hard
Task Description (or 'q' to cancel): Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays.
Interview created with ID: 6
Task assigned to interview successfully.
```

Rysunek 7: Tworzenie zadania rekrutacyjnego.

- **5. Show Unrated Interviews** – raport prezentujący rozmowy kwalifikacyjne, które jeszcze nie zostały ocenione.

```
10. Show All Notes
11. Exit
Choose an option (1-11): 5

--- Unrated Interviews ---
InterviewId: 6, CandidateId: 9, Name: Pawel Sipko, Interviewer: Yuliya Zviarko, Date: 2025-06-10
--- Recruitment Console Menu ---
```

Rysunek 8: Nieocenione rozmowy kwalifikacyjne.

- **6. Evaluate Interview** – proces oceny rozmowy kwalifikacyjnej podczas jej trwania. Wybrana jest osoba odpowiedzialna za ocenę, która wprowadza ocenę kandydata na podstawie ID rozmowy. Ocena jest w skali od 1 do 5. W przypadku podania wartości spoza tego zakresu aplikacja informuje o błędzie. Finalny werdykt nie jest ustalany na tym etapie (służy do tego opcja numer 7).

```

Choose an option (1-11): 6
--- Unrated Interviews ---
InterviewId: 6, CandidateId: 9, Name: Pawel Sipko, Interviewer: Yuliya Zviarko, Date: 2025-06-10
Interview ID (or 'q' to cancel): 6
Evaluator (or 'q' to cancel): Yuliya Zviarko
Score (1-5) (or 'q' to cancel): 5
Interview evaluated.
Recruitment Console Menu

```

Rysunek 9: Ocena rozmowy kwalifikacyjnej.

- **7. Add Final Verdict** – dodanie ostatecznej decyzji dotyczącej rozmowy kwalifikacyjnej podanej przez jej ID. Możliwe wartości to *Accepted* lub *Rejected*.

```

11. Exit
Choose an option (1-11): 7
Enter Interview ID (or 'q' to cancel): 6
Enter final verdict (Accepted / Rejected) (or 'q' to cancel): Accepted
Final verdict added and interview marked as evaluated.
Recruitment Console Menu

```

Rysunek 10: Dodawanie ostatecznego werdyktu.

- **8. Show Candidate Decision** – wyświetlenie pełnych informacji o kandydacie wraz z jego ostatecznym wynikiem rekrutacji.

```

8. Show Candidate Decision
9. Add Note to Candidate
10. Show All Notes
11. Exit
Choose an option (1-11): 8

Candidates:
[9] Pawel Sipko - sipko@gmail.com
[10] Mirosław Kolodziej - kolmir@gmail.com
[11] Patrycja Gorka - gorpar@test.com
[12] Adam Laba - laba@onet.pl
[13] Patryk Tarkowski - patryyyyk@gmail.com
[14] Krzysztof Nowak - krzysznow@gmail.com
[15] Michał Szuplakow - szupmich@agh.edu.pl
[16] Bartosz Rybak - ryba@gmail.com
[17] Kamil Jagodzinski - kjag@agh.edu.pl
[19] Natalia Tomaszewska - tomnat@agh.edu.pl
[21] Anna Dudowska - dudka@com.pl
[23] Magdalena Szewczyk - szewczyk@test.com
[24] Szarlota Ejczison - itscharlibabe@xcx.com
Candidate ID: 9
Candidate: Pawel Sipko (sipko@gmail.com)
Interviewer: Yuliya Zviarko, Date: 2025-06-10
Task: LeetCode hard, Description: Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays.
Evaluator: Yuliya Zviarko, Score: 5
Final Status: Accepted

```

Rysunek 11: Informacje o kandydacie i wynik rekrutacji.

- **9. Add Note to Candidate** – dodanie notatki dla działu HR zawierającej kluczowe uwagi po rozmowie kwalifikacyjnej, takie jak mocne i słabe strony kandydata. Notatki ułatwiają późniejszy wybór spośród kilku wyróżniających się kandydatów.

```

Choose an option (1-11): 9

Candidates:
[9] Pawel Sipko - sipko@gmail.com
[10] Mirosław Kolodziej - kolmir@gmail.com
[11] Patrycja Gorka - gorpar@test.com
[12] Adam Laba - laba@onet.pl
[13] Patryk Tarkowski - patryyyyk@gmail.com
[14] Krzysztof Nowak - krzysznow@gmail.com
[15] Michał Szuplakow - szupmich@agh.edu.pl
[16] Bartosz Rybak - ryba@gmail.com
[17] Kamil Jagodzinski - kjag@agh.edu.pl
[19] Natalia Tomaszewska - tomnat@agh.edu.pl
[21] Anna Dudowska - dudka@com.pl
[23] Magdalena Szewczyk - szewczyk@test.com
[24] Szarlota Ejczison - itscharlibabe@xcx.com
Enter Candidate ID (or 'q' to cancel): 9
Enter note text (or 'q' to cancel): Zadania zostały rozwiązane poprawnie. Kandydat, oprócz dobrej wiedzy z zakresu C++, posiada także dobre umiejętności miękkie - można go zatrudnić.
Note added successfully.

```

Rysunek 12: Dodawanie notatki do profilu kandydata.

- **10. Show All Notes** – raport wszystkich notatek z możliwością zapisania ich do pliku tekstowego.

```

Choose an option (1-11): 10

--- All Candidate Notes ---
[Note ID 3] Candidate ID: 9 | 07.06.2025 15:39 | Zadania zostały rozwiązane poprawnie. Kandydat, oprócz dobrej wiedzy z zakresu C++, posiada także dobre umiejętności miękkie - można go zatrudnić.

Would you like to save the notes to a .txt file? (y/n): y
Notes saved to file: CandidateNotes.txt

```

Rysunek 13: Lista notatek HR.

- **11. Exit** – zakończenie działania programu.

3 Typy danych oraz metody udostępnione w ramach API

Każdy ze stworzonych typów danych (UDT – User Defined Type) zawiera w sobie swoje zmienne prywatne, które przechowują informacje istotne dla danego obiektu. Są one wykorzystywane wewnętrznie w metodach typu oraz udostępniane za pomocą metod typu getter.

- **Candidate:**
 - string name
 - string email
- **TaskItem:**
 - string title
 - string description
- **Interview:**
 - string interviewer
 - string date
- **Evaluation:**
 - string Evaluator
 - int Score
- **RecruitmentSummary:**
 - string candidateName
 - string status
- **HrNote:**
 - string text

– `DateTime createdAt`

Wszystkie UDT posiadają wspólny zestaw metod:

- `ToString()` – zwraca tekstową reprezentację obiektu w postaci rozdzielonej znakiem `|`.
- `Parse(SqlString s)` – tworzy nowy obiekt na podstawie łańcucha znaków.
- `Read(BinaryReader r)` – odczytuje dane obiektu z binarnego strumienia.
- `Write(BinaryWriter w)` – zapisuje dane obiektu do binarnego strumienia.

Dodatkowo, każdy typ posiada specyficzne metody umożliwiające dostęp do wybranych pól:

- **Candidate**

- `ToStringStatic(Candidate)` – statyczna wersja `ToString()`.
- `GetName(Candidate)` – zwraca imię kandydata.
- `GetEmail(Candidate)` – zwraca adres e-mail.
- `IsValidEmail(SqlString)` – sprawdza poprawność formatu adresu e-mail.

- **TaskItem**

- `ToStringStatic(TaskItem)` – statyczna wersja `ToString()`.
- `GetTitle(TaskItem)` – zwraca tytuł zadania.
- `GetDescription(TaskItem)` – zwraca opis zadania.

- **Interview**

- `ToStringStatic(Interview)` – statyczna wersja `ToString()`.
- `GetInterviewer(Interview)` – zwraca imię i nazwisko osoby przeprowadzającej rozmowę.
- `GetDate(Interview)` – zwraca datę rozmowy.

- **Evaluation**

- `ToStringStatic(Evaluation)` – statyczna wersja `ToString()`.
- `GetEvaluator(Evaluation)` – zwraca nazwę osoby oceniającej.
- `GetScore(Evaluation)` – zwraca ocenę (liczbę całkowitą).

- **RecruitmentSummary**

- `ToStringStatic(RecruitmentSummary)` – statyczna wersja `ToString()`.
- `GetCandidateNameSum(RecruitmentSummary)` – zwraca imię kandydata.
- `GetStatusSum(RecruitmentSummary)` – zwraca status rekrutacji (`Accepted`, `Rejected`, `Pending`).

- **HrNote**

- `GetText(HrNote)` – zwraca treść notatki.
- `GetCreatedAt(HrNote)` – zwraca datę utworzenia notatki.

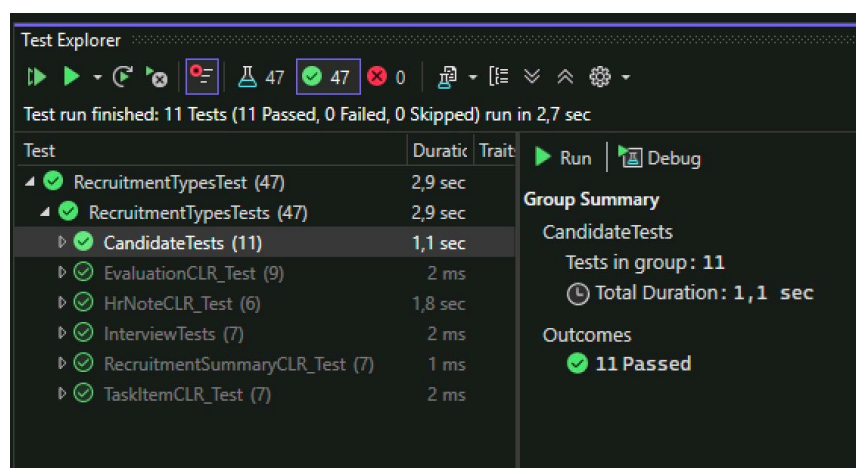
4 Implementacja

Projekt realizowałam na maszynie wirtualnej, udostępnionej w ramach zajęć. Całość została zaimplementowana w trzech osobnych projektach:

- **RecruitmentTypes** – bazowy projekt odpowiedzialny za stworzenie wszystkich UDT (User Defined Types), wymienionych w punkcie 1. W tym projekcie zdefiniowane zostały wszystkie typy danych, co umożliwiło kompilację pliku w formacie .dll. Tak przygotowany assembly przenosiłam do środowiska Microsoft SQL Server, gdzie mogłam wykorzystywać te typy do tworzenia tabel, rozbudowywania funkcjonalności technicznych, definiowania widoków oraz walidacji danych.
- **RecruitmentTypesTest** – projekt przeznaczony do testowania wcześniej napisanych UDT. Dla każdego typu danych przygotowałam osobny zestaw testów jednostkowych, obejmujących wszystkie zdefiniowane metody. Pozwoliło mi to na weryfikację poprawności działania każdej funkcji oraz na wychwycenie ewentualnych błędów we wczesnym etapie prac.
- **RecruitmentTypesConsoleApp** – ostatni z projektów, odpowiadający za integrację logiki napisanej w języku C# z kodem T-SQL. Łączenie z bazą danych realizowałam przy użyciu technologii ADO.NET, co umożliwiło dynamiczną modyfikację zawartości tabel oraz korzystanie z funkcjonalności udostępnionych w SQL Server. Aplikacja zawiera wszystkie niezbędne metody opisane w punkcie 2. Każda z metod wykorzystywała wcześniej przygotowane funkcje w języku T-SQL – w niektórych przypadkach opierałam się na gotowych rozwiązaniach, w innych konieczne było rozszerzenie logiki w C#. Dodatkowo zadbałam o odpowiednią obsługę wyjątków.

5 Prezentacja przeprowadzonych testów jednostkowych

Wszystkie testy jednostkowe, o których mowa w punkcie poprzednim, udało się uruchomić, uzyskując pozytywne wyniki dla każdego przypadku testowego:



Rysunek 14: Wyniki testów jednostkowych.

6 Podsumowanie i wnioski

W trakcie realizacji projektu udało się stworzyć kompletny zestaw typów danych użytkownika (UDT) dla systemu rekrutacyjnego, które zostały skutecznie zaimplementowane, przetestowane i zintegrowane z bazą danych. Dzięki podziałowi na osobne projekty udało się zachować modularność kodu oraz umożliwić jego łatwą rozbudowę i utrzymanie.

W przyszłości planowane jest rozszerzenie biblioteki o dodatkowe funkcjonalności, takie jak bardziej zaawansowane metody wyszukiwania czy edytowanie danych.

7 Kod źródłowy

Całość projektu została załączona w wyznaczonym miejscu na platformie UPEL.

Kod źródłowy jest także dostępny na GitHub pod adresem:

<https://github.com/juliazwierko/WFiIS-BD2-project-RecruitmentSystem>

8 Literatura

- https://upel.agh.edu.pl/pluginfile.php/460230/mod_resource/content/1/BD2_L09_v2022_CLR-Teoria.pdf
- <https://www.youtube.com/watch?v=qlZHI3FTZMo>
- <https://www.youtube.com/watch?v=24ytq8JP2jI&t=586s>