



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: PROGRAMOWANIE

Grupa: SL03

Hlib Hospodarysko w69976

System zarządzania donacjami organów i krwi

Prowadzący: mgr inż. Ewa Żesławska

Praca projektowa. Programowanie Obiektowe C#

Rzeszów 2025

Spis treści

1	Opis założeń projektu	5
1.1	Cele projektu	5
1.2	Wymagania funkcjonalne i нефункционалне	6
1.2.1	Wymagania funkcjonalne	6
1.2.2	Wymagania нефункционалне	6
2	Opis struktury Projektu	7
2.1	Struktura oraz opis techniczny	7
2.1.1	Baza danych	8
2.1.2	Warstwa logiczna	8
2.1.3	Interfejs użytkownika	8
2.2	Wykorzystany język, narzędzia oraz minimalne wymagania sprzętowe	9
2.2.1	Języki	9
2.2.2	Narzędzia	9
2.2.3	Minimalne wymagania sprzętowe	9
2.3	Zarządzanie danymi	9
2.4	Hierarchia klas i opis metod	9
2.4.1	Klasy reprezentujące dane	9
2.4.2	Klasy techniczne	11
2.4.3	Okna główne	12
2.4.4	Okna pomocnicze	12
3	Harmonogram realizacji projektu	13
3.1	Harmonogram	13
3.2	Repozytorium	13
4	Prezentacja warstwy użytkowej	14
5	Podsumowanie	21
	Bibliografia	22
	Spis rysunków	22

Wstęp

Zarządzanie donacjami krwi i organów jest bardzo istotnym elementem systemów opieki zdrowotnej. Dobre zarządzanie tymi procesami ma kluczowe znaczenie nie tylko dla zapewnienia odpowiedniego dostępu do ratujących życie zasobów, ale także dla optymalizacji działań w kontekście organizacyjnym, w którym biorą udział zarówno placówki medyczne, jak i darczyńcy oraz osoby potrzebujące przeszczepów. Problemem, z którym często spotykają się placówki medyczne, jest koordynacja działań związanych z donacjami. Brak odpowiedniego systemu do śledzenia i zarządzania tymi procesami może prowadzić do opóźnień, a także do nieefektywnego wykorzystania dostępnych zasobów. To z kolei ma wpływ na czas oczekiwania na przeszczepienie i na jakość opieki nad pacjentami. Rozwiązaniem jest stworzenie takiego systemu, który zapewni prostotę, przejrzystość i skuteczność w zarządzaniu donacjami krwi i organów.

Rozdział 1

Opis założeń projektu

1.1 Cele projektu

Celem projektu jest stworzenie systemu ułatwiającego zarządzanie donacjami organów i krwi. System ma wspierać skuteczną komunikację dotyczącą przeszczepienia narządów między różnymi jednostkami w tej branży.

Takie narzędzie może znacznie usprawnić przebieg wszystkich działań niezbędnych przed ustaleniem operacji przeszczepienia narządów. Ponadto system będzie mógł przyspieszyć proces podejmowania decyzji dotyczących odpowiednich kombinacji grup krwi i innych czynników niezbędnych dla przeszczepu.

Czas jest jednym z najważniejszych czynników w tego typu operacjach. Dodatkowo brak zunifikowanej bazy informacji o dostępnych organach może prowadzić do krytycznej straty czasu.

Aby rozwiązać ten problem, konieczne jest stworzenie systemu umożliwiającego przechowywanie, dodawanie i modyfikację informacji o dawcach organów, dostępnych narządach i kandydatach na przeszczep.

Realizacja systemu wymaga implementację następujących elementów:

1. Baza danych umożliwiająca przechowywanie informacji o dawcach, kandydatach do przeszczepu, dostępnych narządach, placówkach medycznych oraz ilości dostępnej krwi.
2. Prosty i intuicyjny interfejs użytkownika, który zapewnia szybki i łatwy dostęp do danych.
3. Zasoby techniczne przetwarzające dane podane przez użytkowników na informacje przechowywane w bazie danych.
4. Zasoby walidacji danych
5. Skuteczna komunikacja między elementami bazy danych.

1.2 Wymagania funkcjonalne i нефункционалне

1.2.1 Wymagania funkcjonalne

1. **Zarządzanie danymi dawcy.**
System powinien umożliwiać zarządzanie danymi dawców organów lub krwi.
2. **Zarządzanie danymi biorcy.**
System powinien umożliwiać zarządzanie danymi kandydatów na przeszczep.
3. **Zarządzanie danymi jednostek medycznych.**
System powinien umożliwiać zarządzanie danymi jednostek medycznych.
4. **Zarządzanie danymi poszczególnych donacji.**
System powinien umożliwiać zarządzanie danymi poszczególnych donacji krwi lub organów.
5. **Interfejs użytkownika.**
System powinien posiadać intuicyjny interfejs użytkownika.

1.2.2 Wymagania нефункционалне

1. **Jednorodność.**
Każdy wpis musi być podobny siebie.
2. **Skalowalność**
Dodanie nowych tabeli musi być łatwe.
3. **Przechowywanie historii**
Baza danych musi również przechowywać dane z przeszłości.
4. **Wydajność.**
Czas odpowiedzi systemu na zapytania musi być krótki.
5. **Użyteczność.**
Interfejs użytkownika powinien być intuicyjny.

Rozdział 2

Opis struktury Projektu

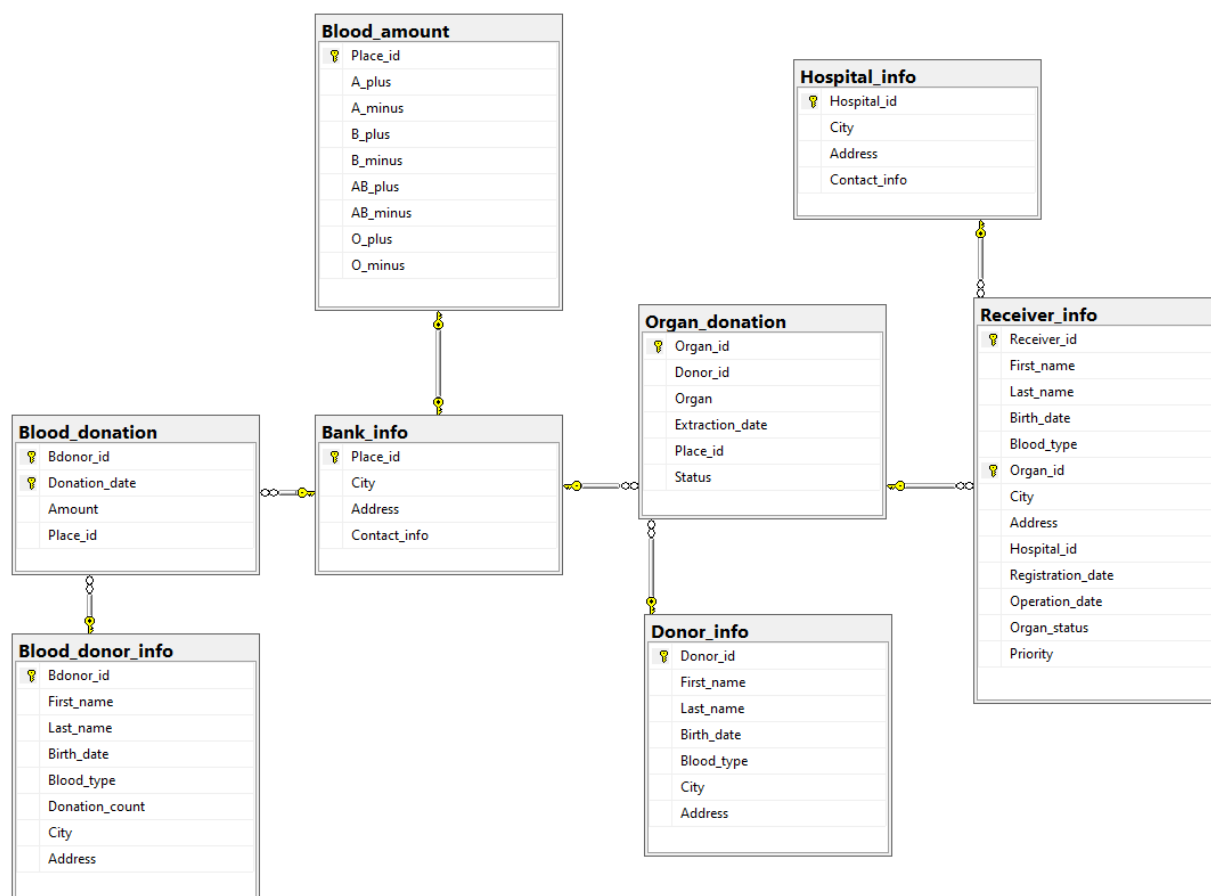
2.1 Struktura oraz opis techniczny

System składa się z trzech głównych komponentów: bazy danych, warstwy logicznej oraz interfejsu użytkownika.

- Baza danych umożliwia przechowywanie oraz dostęp do informacji.
- Warstwa logiczna umożliwia komunikację między interfejsem użytkownika a bazą danych oraz przeprowadza walidację danych.
- Interfejs użytkownika umożliwia łatwe zarządzanie danymi w bazie danych.

2.1.1 Baza danych

Baza danych składa się z ośmiu tabel przechowujących informacje o dawcach krwi i organów, kandydatach na przeszczep, donacjach krwi i organów oraz jednostkach medycznych. Tabele te są powiązane różnorodnymi relacjami, które zapewniają spójność danych. Baza danych zawiera również ograniczenia długości wpisów, wewnętrzny system walidacji danych oraz mechanizm sprawdzania dat, mające na celu zapobieganie wprowadzaniu bezsensownych informacji.



Rysunek 2.1: Diagram ERD bazy danych systemu

2.1.2 Warstwa logiczna

Warstwa logiczna składa się z ośmiu klas obiektów, które stanowią reprezentacje tabel w bazie danych. Klasy te dziedziczą po czterech klasach pomocniczych, zgodnie z zasadami programowania obiektowego. Również system zawiera dwie klasy techniczne, które umożliwiają komunikację z bazą danych oraz walidację danych.

2.1.3 Interfejs użytkownika

Interfejs użytkownika składa się z kilku okien utworzonych na podstawie Windows Forms, podzielonych na trzy warstwy: menu główne; okna poszczególnych kategorii danych, wywoływane za pomocą przycisków w menu głównym; okna pomocnicze, wywoływane za pomocą przycisków w oknach danych.

2.2 Wykorzystany język, narzędzia oraz minimalne wymagania sprzętowe

2.2.1 Języki

- System został stworzony w języku **C#** z wykorzystaniem frameworka **.NET**
- Baza danych została stworzona przy użyciu języka **SQL**

2.2.2 Narzędzia

- Microsoft Visual Studio. Community Edition 2022
- Microsoft Server Management Studio 19

2.2.3 Minimalne wymagania sprzętowe

- RAM: 2Gb
- Procesor: 2GHz
- OS: Windows 10
- Dysk: 100mb wolnego miejsca

2.3 Zarządzanie danymi

Dane w systemie są przechowywane na lokalnym serwerze SQL, utworzonym za pomocą MSMS, w tabelach przedstawionych na diagramie powyżej. Operacje na danych są realizowane przy użyciu klasy **DatabaseManager.cs**, która korzysta z języka LINQ do manipulacji danymi.

2.4 Hierarchia klas i opis metod

2.4.1 Klasy reprezentujące dane

1. **SqlObject.cs**

Klasa wirtualna. Posiada właściwość `parameters`, która jest listą parametrów SQL przypisanych do każdego obiektu z klas rzeczywistych.

2. **Person.cs**

Klasa wirtualna dziedzicząca po `SqlObject`. Posiada właściwości wspólne dla dawców organów, dawców krwi oraz kandydatów: `Id`, `Imie`, `Nazwisko`, `Data urodzenia`, `Grupa krwi`, `Miasto` i `Adress zamieszkania`.

3. **Donation.cs**

Klasa wirtualna dziedzicząca po `SqlObject`. Posiada właściwości wspólne dla donacji krwi i organów: `Id`, `Id miejsca przechowywania`, `Data donacji`.

4. **Place.cs**

Klasa wirtualna dziedzicząca po `SqlObject`. Posiada właściwości wspólne dla jednostek medycznych: `Id`, `Informacja kontaktowa`, `Miasto` i `adres`.

5. **OrganDonor.cs**
Klasa publiczna dziedzicząca po klasie `Person`. Poza właściwościami odziedziczonymi, nie posiada żadnych unikalnych.
6. **BloodDonor.cs**
Klasa publiczna dziedzicząca po klasie `Person`. Poza właściwościami odziedziczonymi, posiada właściwości unikalne: Liczba donacji.
7. **BloodDonor.cs**
Klasa publiczna dziedzicząca po klasie `Person`. Poza właściwościami odziedziczonymi, posiada właściwości unikalne: Liczba donacji.
8. **Receiver.cs**
Klasa publiczna dziedzicząca po klasie `Person`. Poza właściwościami odziedziczonymi, posiada właściwości unikalne: Id organu, Id szpitala, Datę rejestracji, Datę operacji, Status organu, Priorytet.
9. **OrganDonation.cs**
Klasa publiczna dziedzicząca po klasie `Donation`. Poza właściwościami odziedziczonymi, posiada właściwości unikalne: Id dawcy, Status, Nazwa.
10. **BloodDonation.cs**
Klasa publiczna dziedzicząca po klasie `Donation`. Poza właściwościami odziedziczonymi, posiada właściwości unikalne: Ilość.
11. **Hospital.cs**
Klasa publiczna dziedzicząca po klasie `Place`. Poza właściwościami odziedziczonymi, nie posiada żadnych unikalnych.
12. **Bank.cs**
Klasa publiczna dziedzicząca po klasie `Place`. Poza właściwościami odziedziczonymi, nie posiada żadnych unikalnych.
13. **BankBloodAmount.cs**
Klasa publiczna dziedzicząca po klasie `SqlObject`. Poza właściwościami odziedziczonymi, posiada właściwości unikalne: Id banku krwi oraz osiem właściwości reprezentujących grupy krwi.

Każda klasa publiczna posiada dwa konstruktory.

Pierwszy konstruktor przyjmuje zmienne, których zawartość zostaje przypisana do odpowiednich właściwości obiektu. Jest wykorzystywany do tworzenia reprezentacji wpisu z bazy danych, który został wygenerowany wewnątrz kodu. W ciele tego konstruktora zawartość właściwości jest przypisywana do listy parametrów SQL `parameters`.

Drugi konstruktor przyjmuje słownik, w którym kluczem jest parametr SQL, a wartością obiekt reprezentujący dane z bazy danych. Konstruktor ten służy do tworzenia reprezentacji wpisu z bazy danych, który został wygenerowany zewnątrz kodu. ciele tego konstruktora zawartość właściwości jest przypisywana na podstawie zawartości słownika.

2.4.2 Klasy techniczne

1. DatabaseManager.cs

Klasa statyczna implementująca operacje CRUD z wykorzystaniem języka LINQ. Zawiera następujące metody:

- `Create()`
Metoda przyjmuje obiekt klasy `SqlObject` i zapisuje jego informacje na podstawie właściwości `parameters`.
- `Read()`
Metoda przyjmuje numer tabeli i zwraca listę, której elementy są słownikami wykorzystywanymi do tworzenia obiektów.
- `Update()`
Metoda przyjmuje numer tabeli, Id wiersza, parametr do zmiany i nową informację.
- `Delete()`
Metoda przyjmuje numer tabeli i Id wiersza do usunięcia.
- `Search()`
Metoda przyjmuje numer tabeli, parametr do wyszukiwania i informację do wyszukiwania. Zwraca listę, której elementy są słownikami wykorzystywanymi do tworzenia obiektów.
- `MaxID()`
Metoda przyjmuje numer tabeli i zwraca największy numer Id.
- `GetCount()`
Metoda przyjmuje numer id i zwraca liczbę donacji dawcy krwi o podanym Id.
- `GetTable()`
Metoda zwraca trzy ciągi znaków o różnych wartościach, które ułatwiają pracę z językiem LINQ. Jest to metoda prywatna, wykorzystywana jedynie w metodach opisanych powyżej. Została przeciążona w sposób umożliwiający przyjęcie obiektu klasy `SqlObject` lub numeru tabeli.

2. ValidationHelper.cs

Klasa statyczna ułatwiająca walidację danych. Zawiera następujące metody:

- `IsValidPhone()`
Metoda do walidacji numeru telefonu.
- `IsValidName()`
Metoda do walidacji imienia lub nazwiska.
- `IsValidID()`
Metoda do walidacji numeru Id.
- `IsValidDate()`
Metoda do walidacji daty.
- `IsValidStatus()`
Metoda do walidacji statusu organu.
- `IsValidType()`
Metoda do walidacji grupy krwi.
- `IsValidNumber()`
Metoda do walidacji liczby.

2.4.3 Okna główne

1. **frmMain.cs**

Okno zawierające przyciski otwierające inne okna.

2. **frmOrgans.cs**

Okno wyświetlające listę organów oraz przyciski umożliwiające otwieranie okien do manipulacji danymi. Zawiera również przyciski umożliwiające filtrowanie organów według ich statusu.

3. **frmDonors.cs**

Okno wyświetlające listę dawców organów oraz przyciski umożliwiające otwieranie okien do manipulacji danymi.

4. **frmReceipients.cs**

Okno wyświetlające listę kandydatów na przeszczep oraz przyciski umożliwiające otwieranie okien do manipulacji danymi.

5. **frmBDonors.cs**

Okno wyświetlające dawców oraz przyciski umożliwiające otwieranie okien do manipulacji danymi. Zawiera również przycisk, który zwiększa ilość donacji.

6. **frmPlaces.cs**

Okno wyświetlające jednostki medyczne oraz przyciski umożliwiające otwieranie okien do manipulacji danymi.

7. **frmAmount.cs**

Okno wyświetlające ilość krwi różnych grup oraz przycisk do manipulacji danymi.

2.4.4 Okna pomocnicze

1. **frm__Add.cs**

Okna pobierające dane w zależności od rodzaju wpisu. Każde z takich okien weryfikuje dane przy użyciu klasy `ValidationHelper`.

Dodanie wpisów do tabeli dawców organów oraz kandydatów na przeszczep skutkuje utworzeniem wpisów w tabeli donacji organów ze statusem "Dostępne" lub "Oczekiwane".

Dodanie wpisów do tabeli Banków krwi skutkuje utworzeniem wpisu w tabeli Ilości krwi.

2. **frm__Add.cs**

Okna usuwające wpisy w zależności od rodzaju wpisu.

3. **frm__Update.cs**

Okna zmieniające dane w zależności od rodzaju wpisu. Każde z takich okien weryfikuje dane przy użyciu klasy `ValidationHelper`.

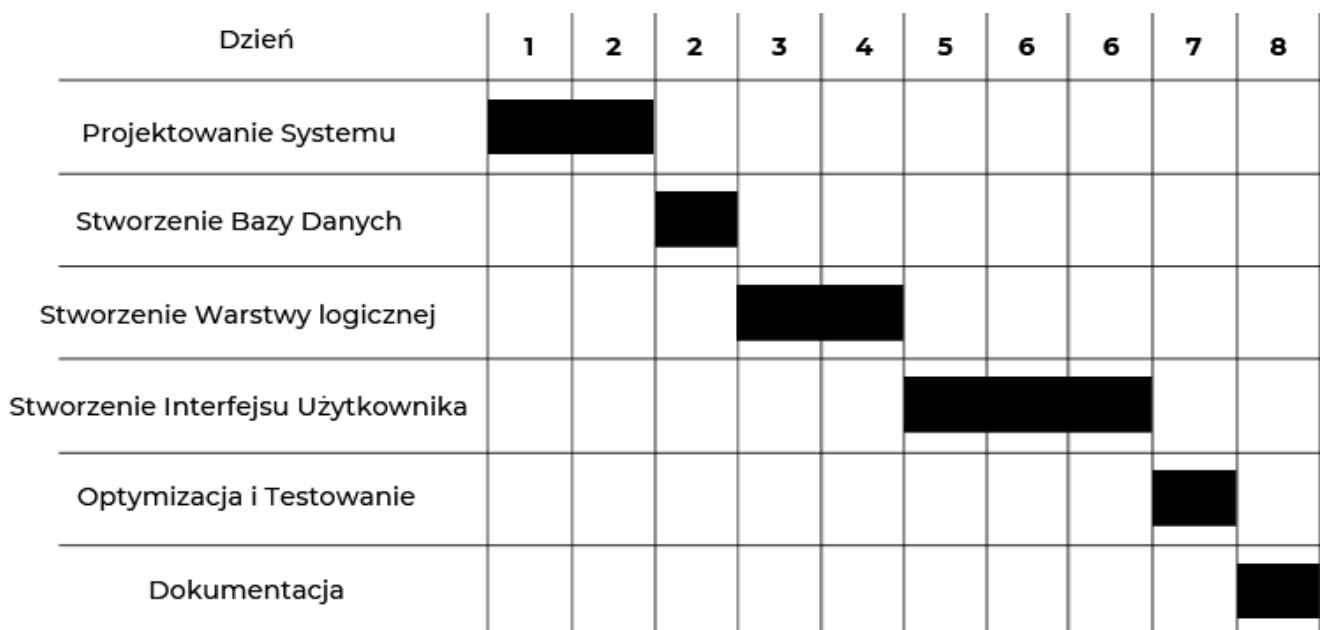
4. **frmBDonorAddDonation.cs**

Okno pobierające dane dotyczące dawki krwi. Zwiększa liczbę dawek odpowiedniego dawcy o 1 oraz aktualizuje ilość krwi danej grupy w tabeli banków krwi.

Rozdział 3

Harmonogram realizacji projektu

3.1 Harmonogram



Rysunek 3.1: Diagram Gantta

3.2 Repozytorium

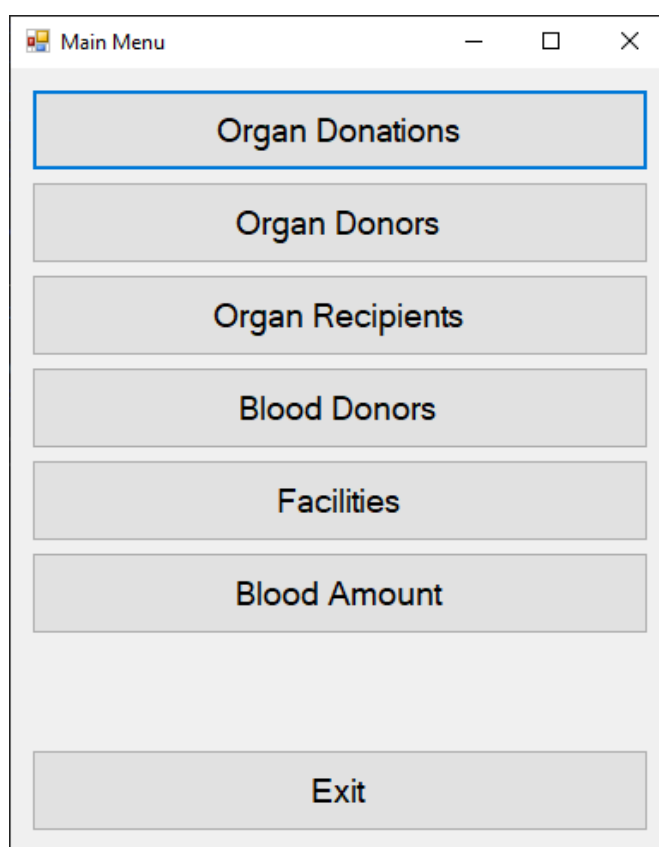
Wszystkie pliki źródłowe zostały zamieszczone w repozytorium, dostępnym pod linkiem:
https://github.com/iSweetyBread/System_for_Management_of_Organ_Donations

Pliki źródłowe obejmują: pliki systemowe, skrypty SQL do tworzenia tabel i wypełniania ich danymi oraz pliki dokumentacji.

Rozdział 4

Prezentacja warstwy użytkowej

Po włączeniu aplikacji użytkownik widzi główne menu.



Rysunek 4.1: Główne menu

Po kliknięciu jednego z głównych przycisków użytkownik zobaczy okna wyświetlające dane.

The screenshot shows a window titled "frmOrgans" with a table of organ donations. The table has columns: Organ ID, Organ Name, Organ Status, Donor ID, Donation Date, and Place ID. The first row is highlighted. Below the table are three buttons: "Available Organs", "Used Organs", and "Awaited Organs". At the bottom are three buttons: "Add Entry", "Remove Entry", and "Update Entry".

Organ ID	Organ Name	Organ Status	Donor ID	Donation Date	Place ID
2	Pluco	Dostępne	2	01.11.2023	2
6	Rogówka	Dostępne	6	30.10.2023	5
12	Watroba	Dostępne	12	05.06.2023	4
15	Rogówka	Dostępne	15	20.08.2023	1
16	Jelito	Dostępne	16	10.10.2023	1
18	Kosc	Dostępne	18	01.01.2024	3
19	Serce	Dostępne	19	21.11.2023	4
21	Watroba	Dostępne	1	30.09.2023	2
24	Rogówka	Dostępne	4	15.05.2023	5
25	Jelito	Dostępne	5	28.01.2024	5

Rysunek 4.2: Organ Donations

The screenshot shows a window titled "frmDonors" with a table of organ donors. The table has columns: Donor ID, First Name, Last Name, Birth Date, Blood Type, City, and Address. The first row is highlighted. Below the table are three buttons: "Add Entry", "Remove Entry", and "Update Entry".

Donor ID	First Name	Last Name	Birth Date	Blood Type	City	Address
1	Jan	Kowalski	12.03.1985	A+	Warszawa	Ul. Marszałkowsk...
2	Anna	Nowak	08.07.1990	B+	Kraków	Ul. Grodzka 12
3	Piotr	Wisniewski	20.11.1978	AB-	Gdansk	Ul. Długa 25
4	Katarzyna	Wójcik	15.06.1989	O+	Wroclaw	Ul. Swidnicka 7
5	Michal	Kaminski	22.01.1983	A-	Poznan	Ul. Półwiejska 19
6	Agnieszka	Lewandowska	30.12.1995	B-	Lódz	Ul. Piotrkowska 38
7	Tomasz	Zielinski	10.09.1987	O-	Lublin	Ul. Krakowskie P...
8	Magdalena	Szymanska	17.02.1992	AB+	Katowice	Ul. Mariacka 60
9	Marcin	Dabrowski	14.08.1976	A+	Bydgoszcz	Ul. Gdanska 30
10	Ewa	Pawlak	18.04.1984	B+	Szczecin	Ul. Wojska Polski...
11	Rafal	Kozłowski	09.03.1993	O+	Rzeszów	Ul. 3 Maja 14

Rysunek 4.3: Organ Donors

frmReipients

	Receipient ID	First Name	Last Name	Birth Date	Blood Type	Organ ID	Organ Status	Priority	Operation Date
▶	1	Jan	Rutkowski	15.03.1975	A+	1	Przyjete	Wysoki	01.03.2023
	2	Anna	Czajka	20.07.1988	B+	5	Przyjete	Sredni	15.05.2023
	3	Piotr	Górski	25.11.1990	AB-	3	Oczekiwane	Niski	
	4	Maria	Sobczak	18.06.1982	O+	4	Przyjete	Wysoki	10.07.2023
	5	Katarzyna	Wójcik	30.01.1995	A-	17	Oczekiwane	Sredni	
	6	Tomasz	Kowal	15.02.1985	O-	20	Przyjete	Wysoki	01.09.2023
	7	Agnieszka	Zelinska	09.10.1993	AB+	13	Oczekiwane	Niski	
	8	Marek	Wisniewski	18.04.1978	B-	9	Przyjete	Sredni	20.10.2023
	9	Beata	Kowalska	05.12.1989	A+	11	Oczekiwane	Wysoki	15.11.2023
	10	Rafal	Nowak	12.05.1992	B+	10	Oczekiwane	Sredni	

Add Entry Remove Entry Update Entry

Rysunek 4.4: Organ Recipients

frmBDonors

	Donor ID	First Name	Last Name	Birth Date	Blood Type	Donation Count
▶	1	Jan	Kowalski	12.03.1985	A+	5
	2	Anna	Nowak	08.07.1990	B+	8
	3	Piotr	Wisniewski	20.11.1978	AB-	3
	4	Katarzyna	Wójcik	15.06.1989	O+	6
	5	Marek	Górski	12.05.1987	A-	7
	6	Ewa	Zawadzka	20.02.1995	B-	4
	7	Pawel	Wiatr	08.08.1980	O+	10
	8	Magdalena	Sawicka	17.11.1983	AB+	2
	9	Grzegorz	Nowicki	19.09.1991	A+	5
	10	Natalka	Gonczarczyk	05.03.1999	B-	6

Add Entry Remove Entry Update Entry

Add Donation

Rysunek 4.5: Blood Donors

frmPlaces

	Hospital ID	City	Address	Contact Info
▶	1	Warszawa	Ul. Banacha 1a	+48 22 599 1000
	2	Kraków	Ul. Kopernika 36	+48 12 424 7000
	3	Gdansk	Ul. Kliniczna 1	+48 58 349 3000
	4	Wroclaw	Ul. Borowska 213	+48 71 733 2000
	5	Poznan	Ul. Przybyszewsk...	+48 61 869 1000

	Bank ID	City	Address	Contact Info
▶	1	Warszawa	Ul. Marszałkowsk...	+48 22 123 4567
	2	Kraków	Ul. Grodzka 12	+48 12 345 6789
	3	Gdansk	Ul. Długa 25	+48 58 987 6543
	4	Wroclaw	Ul. Swidnicka 7	+48 71 246 1357
	5	Poznan	Ul. Półwiejska 19	+48 61 543 2190

Add Facility Remove Facility Update Facility

Rysunek 4.6: Facilities

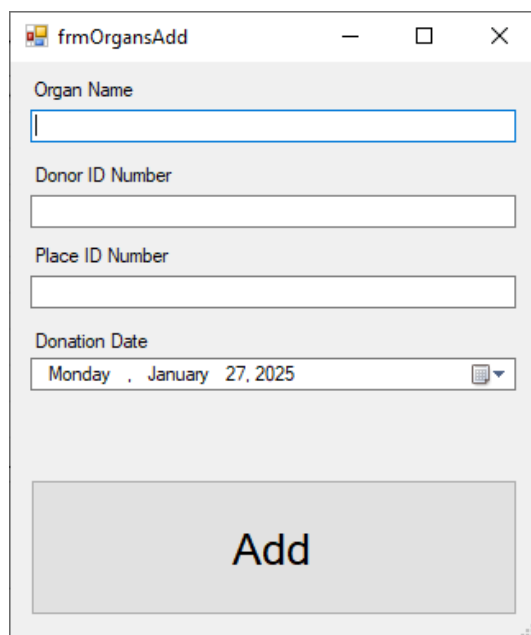
frmAmount

	Bank ID	A+	A-	B+	B-	AB+	AB-	O+	O-
▶	1	120	50	100	40	30	10	200	70
	2	150	60	80	35	40	15	220	90
	3	110	45	90	50	25	12	180	60
	4	140	55	85	30	35	8	190	80
	5	130	65	75	45	20	14	210	75

Update Amount

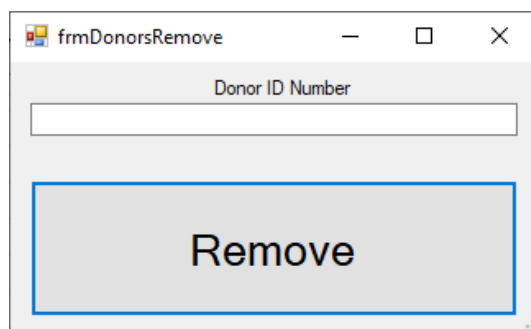
Rysunek 4.7: Blood Amount

Po kliknięciu przycisków manipulacji danymi użytkownik zobaczy okna pobierające dane.



The screenshot shows a Windows-style window titled "frmOrgansAdd". It contains four input fields: "Organ Name", "Donor ID Number", "Place ID Number", and "Donation Date". The "Donation Date" field is a date picker showing "Monday, January 27, 2025". Below these fields is a large grey button with the text "Add".

Rysunek 4.8: Add Organ Donations



The screenshot shows a Windows-style window titled "frmDonorsRemove". It contains a single input field labeled "Donor ID Number". Below this field is a large grey button with the text "Remove".

Rysunek 4.9: Remove Organ Donors

frmReceipientsUpdate

Organ Receptient ID Number

New Data

☐ First Name
☐ Last Name
☐ Birth Date
☐ Blood Type
☐ Organ Status
☐ Priority
☐ Operation Date

Update

Rysunek 4.10: Update Organ Recipients

frmBDonorAddDonation

Blood Donor ID Number

Amount (ml)

Donation Place ID Number

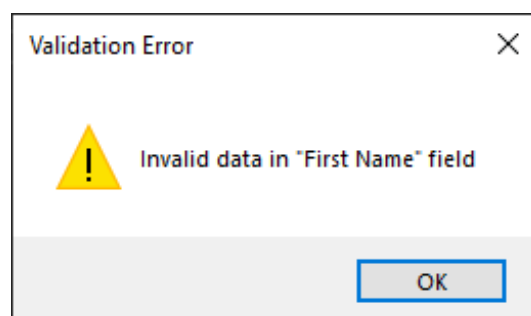
Donation Date

Monday, January 27, 2025

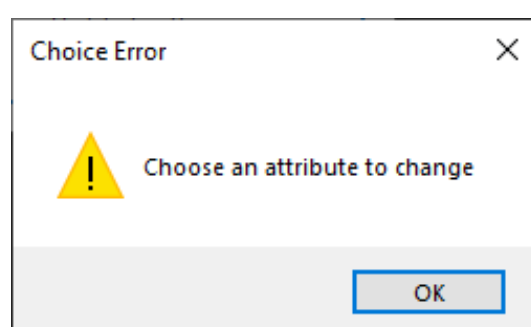
Add

Rysunek 4.11: Add Blood Donation

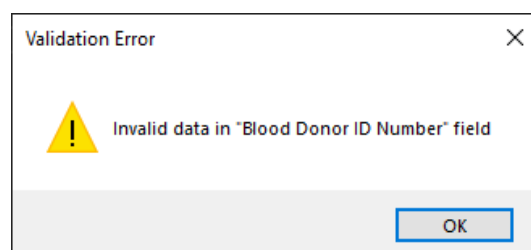
Po wprowadzeniu niepoprawnych danych użytkownik zobaczy okna ostrzegawcze.



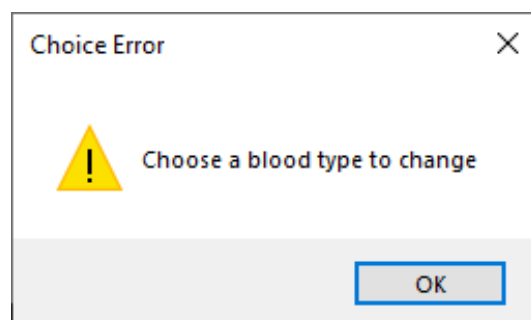
Rysunek 4.12: Niepoprawne dane w pole Imienia



Rysunek 4.13: Niewybrane dane do zmiany



Rysunek 4.14: Niepoprawne dane w pole numeru Id



Rysunek 4.15: Niewybrana grupa do zmiany

Rozdział 5

Podsumowanie

W wyniku pracy nad projektem zostało stworzone proste narzędzie do uporządkowania, przechowywania i zmiany różnych informacji dotyczącej branży przeszczepienia narządów i donacji krwi.

Wszystkie pliki źródłowe są dostępne pod linkiem:

https://github.com/iSweetyBread/System_for_Management_of_Organ_Donations

Potencjalny rozwój:

- Implementacja systemu kontroli dostępu dla pracowników o zróżnicowanych poziomach uprawnień.
- Implementacja narzędzia do porównywania organów z kandydatami na przeszczep zgodnie z kompatybilnością grup krwi.
- Implementacja dostępu online do bazy danych.
- Implementacja obsługi wielu instancji aplikacji.

Bibliografia

- [1] <https://stackoverflow.com/>
- [2] <https://www.w3schools.com/>
- [3] <https://learn.microsoft.com/>

Spis rysunków

2.1	Diagram ERD bazy danych systemu	8
3.1	Diagram Gantt	13
4.1	Główne menu	14
4.2	Organ Donations	15
4.3	Organ Donors	15
4.4	Organ Recipients	16
4.5	Blood Donors	16
4.6	Facilities	16
4.7	Blood Amount	17
4.8	Add Organ Donations	18
4.9	Remove Organ Donors	18
4.10	Update Organ Recipients	19
4.11	Add Blood Donation	19
4.12	Niepoprawne dane w pole Imienia	20
4.13	Niewybrane dane do zmiany	20
4.14	Niepoprawne dane w pole numeru Id	20
4.15	Niewybrana grupa do zmiany	20