



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

## **KOLEGIUM INFORMATYKI STOSOWANEJ**

**Kierunek: INFORMATYKA**

**Specjalność: Programowanie**

Patryk Pieniążek  
Nr albumu studenta 67174

### ***Biblioteka w asp.net (MVC)***

Prowadzący: mgr inż. Łukasz Piechocki

## **Projekt**

**Rzeszów 2024**

# Spis treści

<b>1</b>	<b>Opisy, założenia i wymagania projektu</b>	<b>3</b>
1.1	Założenia i cele projektu . . . . .	3
1.2	Główne założenia . . . . .	3
1.3	Stos technologiczny . . . . .	3
1.4	Instrukcja uruchomienia aplikacji . . . . .	3
<b>2</b>	<b>Część teoretyczna</b>	<b>4</b>
2.1	Diagram ERD . . . . .	4
2.2	Diagram przypadków użycia . . . . .	4
2.3	Kluczowe elementy backendu . . . . .	5
<b>3</b>	<b>Część praktyczna</b>	<b>8</b>
3.1	Interfejs aplikacji . . . . .	8
3.2	Test połączenia z bazą danych . . . . .	11
	<b>Bibliografia</b>	<b>15</b>
	<b>Spis rysunków</b>	<b>15</b>

# Rozdział 1

## Opisy, założenia i wymagania projektu

### 1.1 Założenia i cele projektu

Repozytorium projektu znajduje się na stronie: "<https://github.com/ppieniazek/st2>".

Celem projektu jest stworzenie aplikacji asp.net zarządzającej wypożyczaniem książek w bibliotece.

### 1.2 Główne założenia

- Zarządzanie książkami (CRUD)
- Zarządzanie autorami (CRUD)
- Zarządzanie klientami (CRUD)
- Zarządzanie wypożyczeniami (CRUD)

### 1.3 Stos technologiczny

Projekt wykorzystuje .NET w wersji 8.0 w połączeniu z ASP.NET (model-view-controller), został stworzony w środowisku Visual Studio 2022 Community Edition. Serwer SQL użyty w projekcie to lokalny Microsoft SQL Server Express Edition (64-bit), który zawiera się w środku Visual Studio.

### 1.4 Instrukcja uruchomienia aplikacji

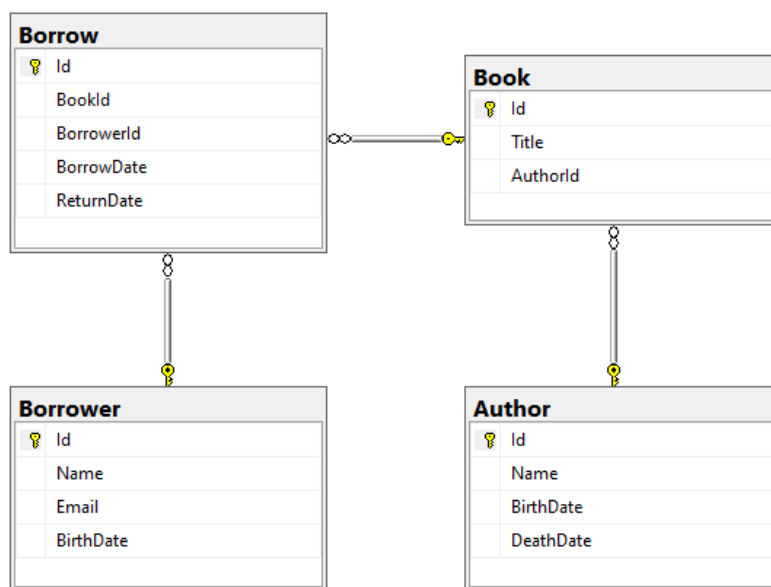
Aby poprawnie pobrać i uruchomić aplikację trzeba wykonać poniższe kroki:

- Pobrać i zainstalować Visual Studio 2022 oraz git z oficjalnej strony.
- W terminalu wykonać komendę "git clone <https://github.com/ppieniazek/st2>.git"
- Uruchomić plik "st2/Library/Library.sln"
- Jeśli nie otworzyła się domyślnie to uruchomić konsolę menadżera pakietów NuGet (Narzędzia -> Menadżer pakietów NuGet -> Konsola menadżera pakietów)
- W konsoli wykonać komendę "Update-Database"
- Uruchomić program (CTRL + F5)

# Rozdział 2

## Część teoretyczna

### 2.1 Diagram ERD

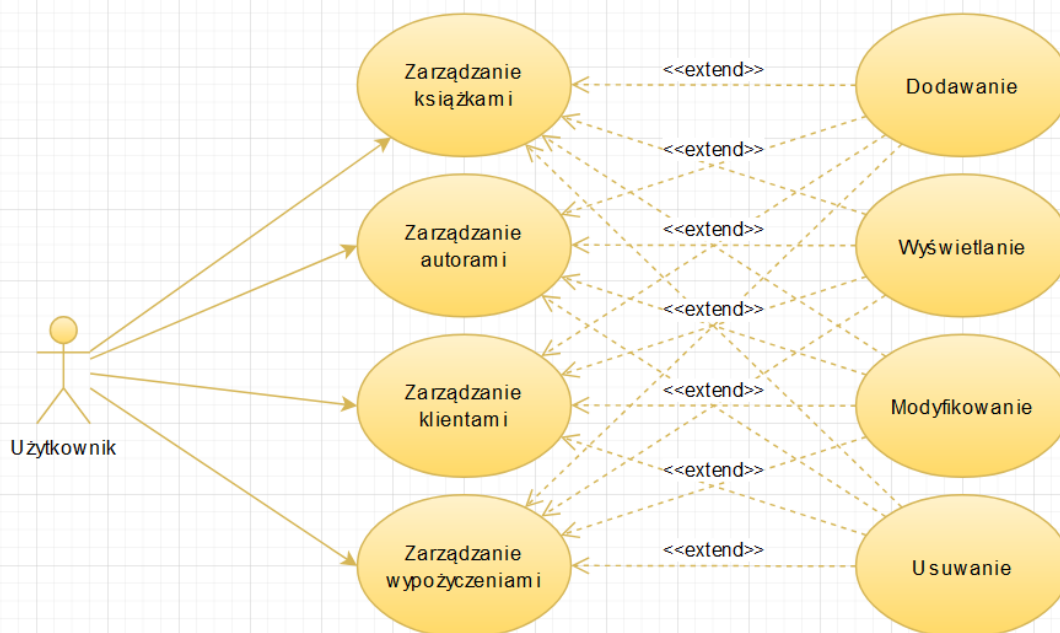


Rysunek 2.1: Diagram ERD

Baza danych zawiera 4 tabele:

- Book - informacje o książkach
- Author - informacje o autorach książek
- Borrower - informacje o klientach wypożyczających książki
- Borrow - informacje o wypożyczeniach

### 2.2 Diagram przypadków użycia



Rysunek 2.2: Diagram przypadków użycia

## 2.3 Kluczowe elementy backendu

Kluczowymi elementami backendu są wszystkie 4 klasy, które są modelami dla bazy danych oraz Library-Context, który określa strukturę bazy danych.

```

public class Book
{
    [Key]
    public int Id { get; set; }
    public string Title { get; set; }
    [DisplayName("Author")]
    public int? AuthorId { get; set; }

    @references
    public virtual Author? Author { get; set; }
}
  
```

Rysunek 2.3: Model tabeli Book

```

public class Author
{
    [Key]
    13 references
    public int Id { get; set; }
    13 references
    public string Name { get; set; }
    [DataType(DataType.Date)]
    [DisplayName("Date of birth")]
    12 references
    public DateTime BirthDate { get; set; }
    [DataType(DataType.Date)]
    [DisplayName("Date of death")]
    12 references
    public DateTime? DeathDate { get; set; }
}

```

Rysunek 2.4: Model tabeli Author

```

public class Borrow
{
    [Key]
    11 references
    public int Id { get; set; }
    [DisplayName("Book Title")]
    8 references
    public int? BookId { get; set; }
    9 references
    public virtual Book? Book { get; set; }
    [DisplayName("Borrower Name")]
    8 references
    public int? BorrowerId { get; set; }
    9 references
    public virtual Borrower? Borrower { get; set; }
    [DataType(DataType.Date)]
    [DisplayName("Date of borrowing")]
    12 references
    public DateTime BorrowDate { get; set; }
    [DataType(DataType.Date)]
    [DisplayName("Date of return")]
    12 references
    public DateTime? ReturnDate { get; set; }
}

```

Rysunek 2.5: Model tabeli Borrow

```

public class Borrower
{
    [Key]
    public int Id { get; set; }

    public string Name { get; set; }
    [DataType(DataType.EmailAddress)]

    public string Email { get; set; }
    [DataType(DataType.Date)]
    [DisplayName("Date of birth")]

    public DateTime BirthDate { get; set; }
}

```

Rysunek 2.6: Model tabeli Borrower

```
public class LibraryContext : DbContext
{
    public LibraryContext(DbContextOptions<LibraryContext> options)
        : base(options)
    {
    }

    public DbSet<BackEnd.Models.Author> Author { get; set; } = default!;
    public DbSet<BackEnd.Models.Book> Book { get; set; } = default!;
    public DbSet<BackEnd.Models.Borrower> Borrower { get; set; } = default!;
    public DbSet<BackEnd.Models.Borrow> Borrow { get; set; } = default!;
}
```

Rysunek 2.7: Klasa LibraryContext

# Rozdział 3

## Część praktyczna

### 3.1 Interfejs aplikacji

Aplikacja po uruchomieniu wyświetla menu startowe (home), które informuje nas, żeby posługiwać się górnym menu. Każda zakładka odpowiada zarządzaniu daną tabelą, która w pozycji startowej wyświetla wszystkie rekordy i posiada cztery funkcjonalności:

- Tworzenie nowych rekordów w tabeli (link "Create New" w widoku startowym)
- Edytowanie istniejących rekordów w tabeli (link "Edit" przy każdym z rekordów)
- Usuwanie istniejących rekordów z tabeli (link "Delete" przy każdym z rekordów)
- Wyświetlanie szczegółów o rekordzie w tabeli (link "Details" przy każdym z rekordów)

Poniżej umieszczone zostały screenshoty pokrywające powyższą funkcjonalność.



Rysunek 3.1: Widok startowy aplikacji



## Borrows

[Create New](#)

Book	Borrower	Date of borrowing	Date of return	
Python - Podstawy	<a href="mailto:email@test.com">email@test.com</a>	01/07/2024		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Python - Podstawy	<a href="mailto:test2@wp.pl">test2@wp.pl</a>	18/06/2024	25/06/2024	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

Rysunek 3.2: Widok startowy dla wypożyczeń

## Create

Book

Title

Author

Create

[Back to List](#)

Rysunek 3.3: Widok dodawania elementu dla książek

## Details

### Borrower

**Name** Patryk Pieniążek  
**Date of birth** 17/10/2001  
**Email** [email@test.com](mailto:email@test.com)

[Edit](#) | [Back to List](#)

Rysunek 3.4: Widok szczegółów dla klienta

## Delete

Are you sure you want to delete this?

### Author

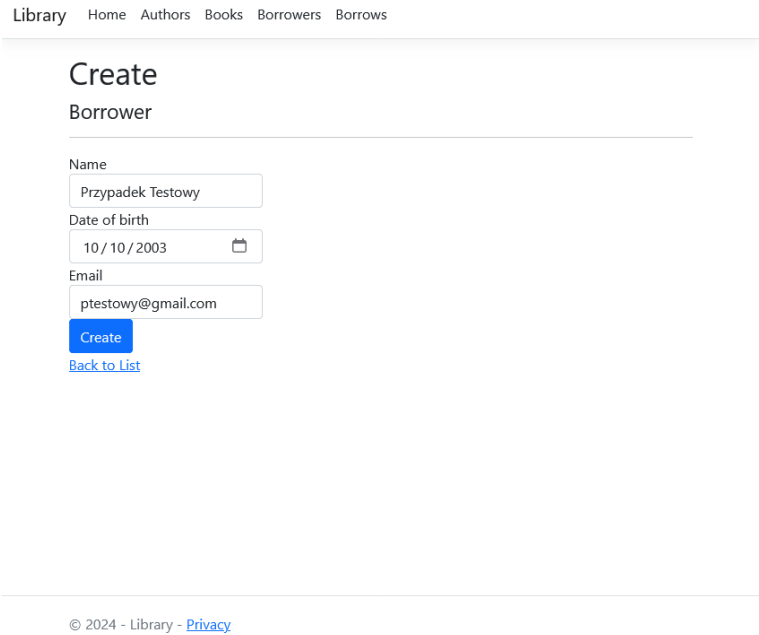
**Name** Adam Mickiewicz  
**Date of birth** 24/06/2024  
**Date of death** 04/07/2024

[Delete](#) | [Back to List](#)

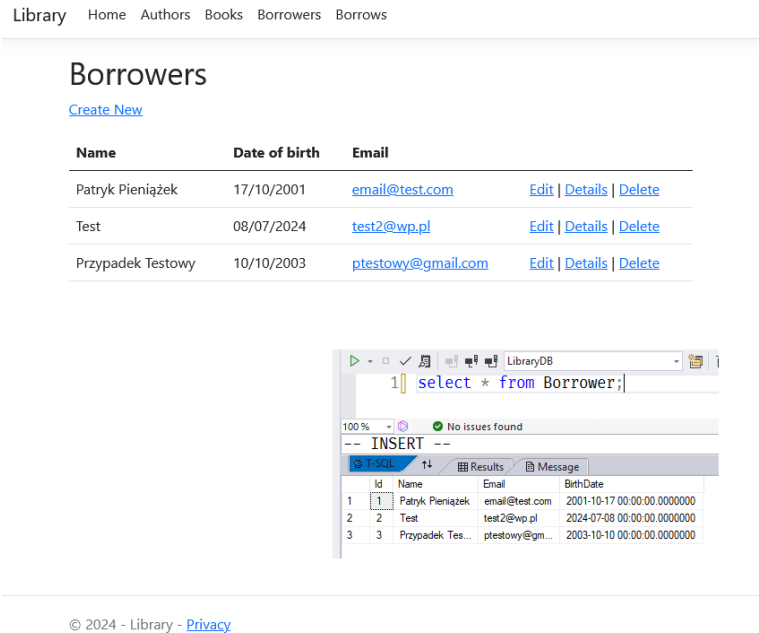
Rysunek 3.5: Widok usuwania elementu dla autorów

### 3.2 Test połączenia z bazą danych

Test polega na utworzeniu nowego klienta, edytowania jego emaila i na końcu jego usunięcia. Po każdej operacji będę sprawdzał zmiany w bazie danych.



Rysunek 3.6: Widok create dla testu



Rysunek 3.7: Sprawdzenie istnienia nowego rekordu w gui i bazie danych

## Edit

## Borrower

Name

Przypadek Testowy

Date of birth

10 / 10 / 2003

Email

emailpozmianie@test.com

Save

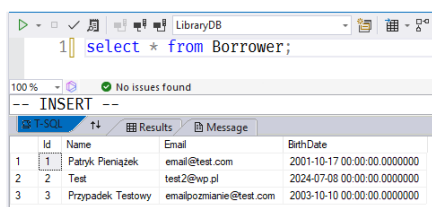
[Back to List](#)

Rysunek 3.8: Widok edit dla testu

## Borrowers

[Create New](#)

Name	Date of birth	Email	
Patryk Pieniążek	17/10/2001	<a href="#">email@test.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Test	08/07/2024	<a href="#">test2@wp.pl</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Przypadek Testowy	10/10/2003	<a href="#">emailpozmianie@test.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>



The screenshot shows a database management interface. At the top, there's a toolbar with various icons. Below it, a text area contains the SQL query: `select * from Borrower;`. Below the query, there's a status bar indicating '100 %' and 'No issues found'. Below that, there's a section labeled 'INSERT' with a table of results. The table has columns: Id, Name, Email, and BirthDate. The results are as follows:

Id	Name	Email	BirthDate
1	Patryk Pieniążek	email@test.com	2001-10-17 00:00:00.0000000
2	Test	test2@wp.pl	2024-07-08 00:00:00.0000000
3	Przypadek Testowy	emailpozmianie@test.com	2003-10-10 00:00:00.0000000

Rysunek 3.9: Sprawdzenie zmiany nowego rekordu w gui i bazie danych

## Delete

Are you sure you want to delete this?

Borrower

**Name** Przekadek Testowy  
**Date of birth** 10/10/2003  
**Email** [emailpozmianie@test.com](mailto:emailpozmianie@test.com)

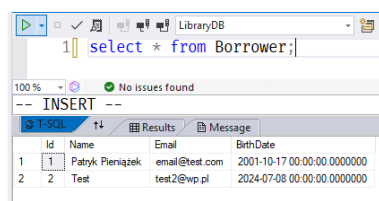
[Delete](#) | [Back to List](#)

Rysunek 3.10: Widok delete dla testu

## Borrowers

[Create New](#)

Name	Date of birth	Email	
Patryk Pieniążek	17/10/2001	<a href="mailto:email@test.com">email@test.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Test	08/07/2024	<a href="mailto:test2@wp.pl">test2@wp.pl</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>



Rysunek 3.11: Sprawdzenie poprawnego usunięcia rekordu w gui i bazie danych

# Podsumowanie

Aplikacja stanowi solidną bazę do dalszego rozwoju i ma potencjał do przekształcenia się z w zaawansowany system zarządzania biblioteką.

# Bibliografia

- [1] <https://learn.microsoft.com/pl-pl/aspnet/mvc/overview/getting-started/introduction/getting-started> - Wprowadzenie do ASP.NET MVC 5 z dnia 01.07.2024
- [2] <https://dotnet.microsoft.com/en-us/learn> - Learning center od Microsoft z dnia 01.07.2024
- [3] <https://learn.microsoft.com/pl-pl/aspnet/core/tutorials/first-mvc-app/working-with-sql> - Połączenie bazy danych z MVC w ASP.NET z dnia 01.07.2024

# Spis rysunków

2.1	Diagram ERD . . . . .	4
2.2	Diagram przypadków użycia . . . . .	5
2.3	Model tabeli Book . . . . .	5
2.4	Model tabeli Author . . . . .	6
2.5	Model tabeli Borrow . . . . .	6
2.6	Model tabeli Borrower . . . . .	6
2.7	Klasa LibraryContext . . . . .	7
3.1	Widok startowy aplikacji . . . . .	8
3.2	Widok startowy dla wypożyczeń . . . . .	9
3.3	Widok dodawania elementu dla książek . . . . .	9
3.4	Widok szczegółów dla klienta . . . . .	10
3.5	Widok usuwania elementu dla autorów . . . . .	10
3.6	Widok create dla testu . . . . .	11
3.7	Sprawdzenie istnienia nowego rekordu w gui i bazie danych . . . . .	11
3.8	Widok edit dla testu . . . . .	12
3.9	Sprawdzenie zmiany nowego rekordu w gui i bazie danych . . . . .	12
3.10	Widok delete dla testu . . . . .	13
3.11	Sprawdzenie poprawnego usunięcia rekordu w gui i bazie danych . . . . .	13