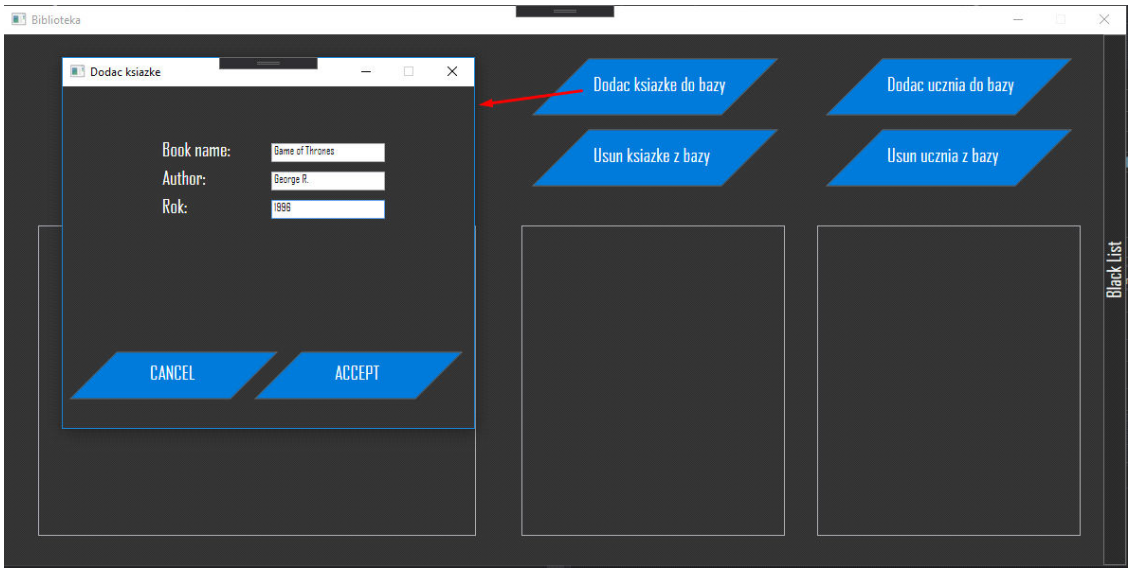
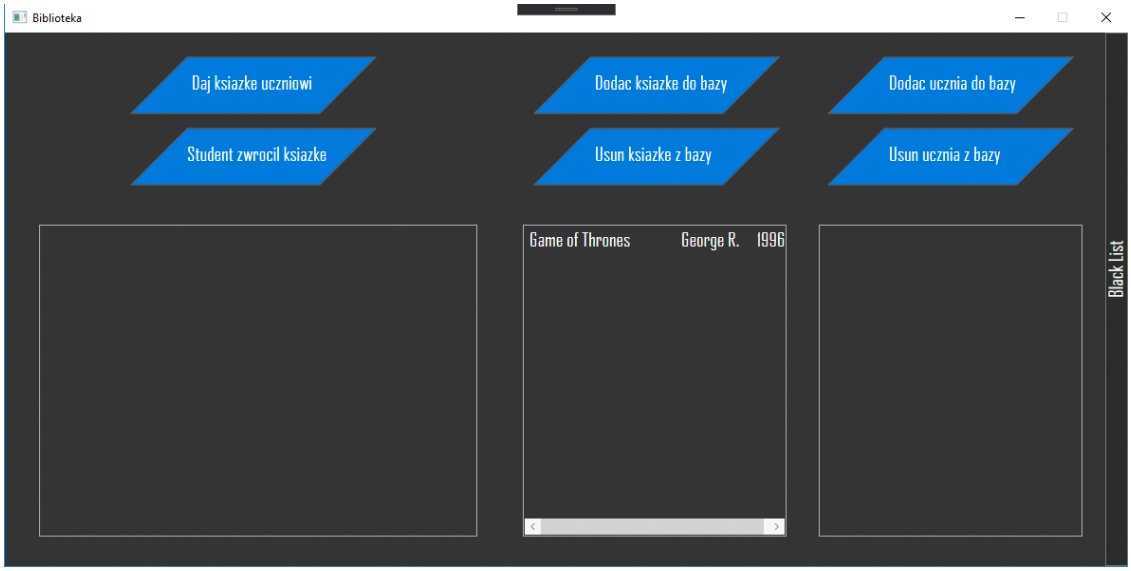
Projekt "Biblioteka"

Artem Levchenko

To jest główne menu programu:

Klikając na "Dodac ksiazke do bazy" otworzy się okno, w którym możemy dodać książkę do bazy danych:

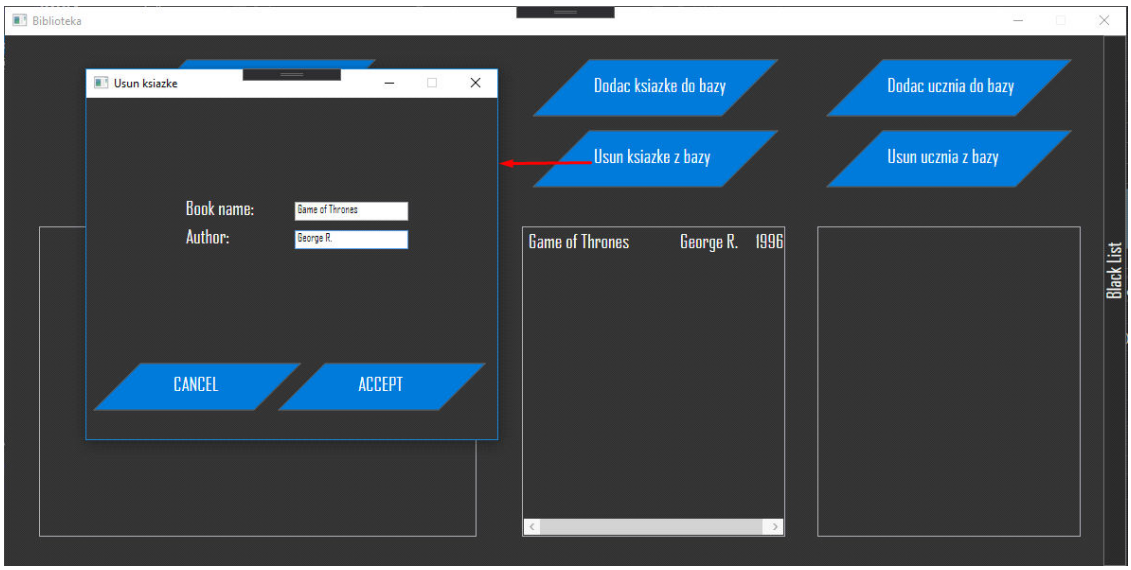




W kodzie jest to zaimplementowane w taki sposób:

```
ссылка: 1
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if(BName.Text != String.Empty & AAuthor.Text != String.Empty & YYear.Text != String.Empty)
    {
        Books book = new Books(BName.Text, AAuthor.Text, YYear.Text);
        BinaryFormatter formatter = new BinaryFormatter();
        FileStream file = new FileStream(@"Books.dat", FileMode.Append);
        formatter.Serialize(file, book);
        file.Close();
        this.Close();
    }
    var Refresh = this.Owner as MainWindow;
    Refresh.Refresh();
}
}
```

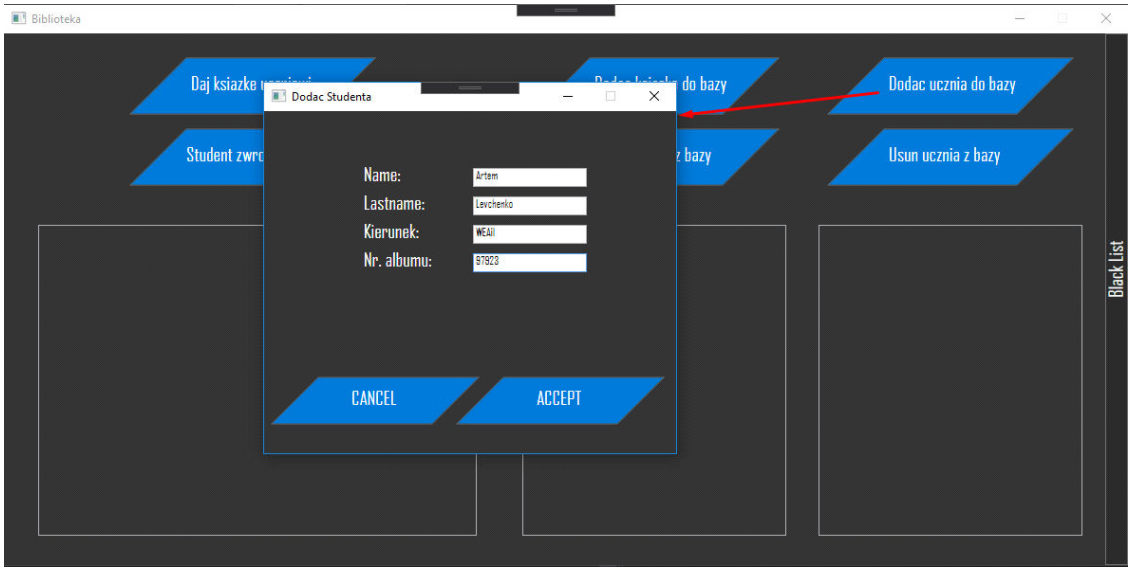
Usuwanie książki z bazy danych wygląda następująco:



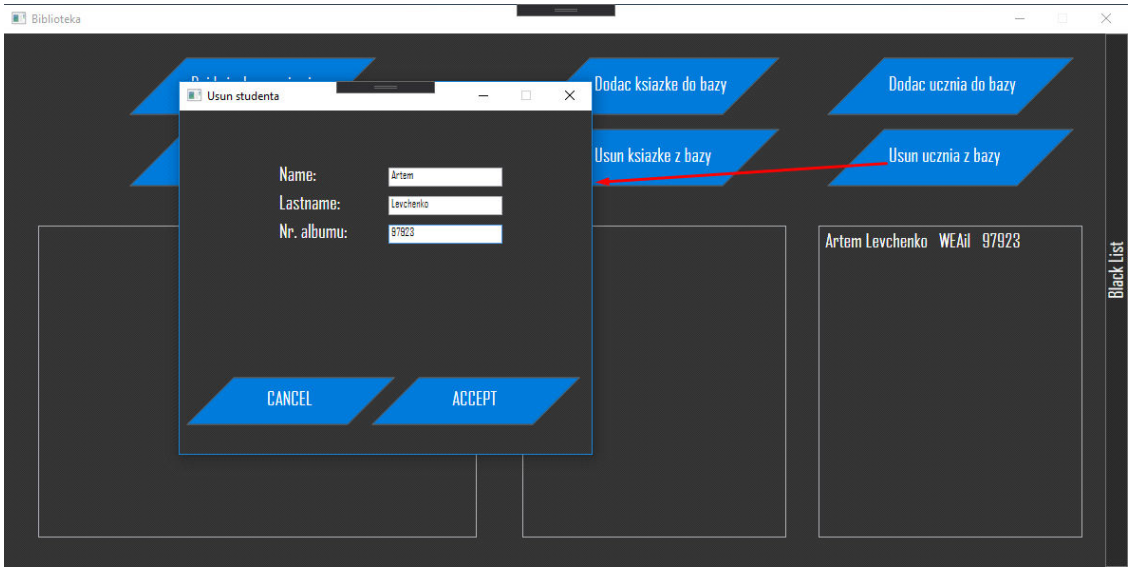

```
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if(BName.Text != String.Empty & AAuthor.Text != string.Empty)
    {
        BinaryFormatter formatter = new BinaryFormatter();
        FileStream fileRead = new FileStream(@"Books.dat", FileMode.OpenOrCreate);
        FileStream fileSave = new FileStream(@"BooksTMP.dat", FileMode.OpenOrCreate);
        BinaryReader read = new BinaryReader(fileRead);
        try
        {
            while (read.PeekChar() >= 0)
            {
                Books book = (Books)formatter.Deserialize(fileRead);
                if(BName.Text == book.NameBook & AAuthor.Text == book.Author) { }
                else
                {
                    formatter.Serialize(fileSave, book);
                }

            }
            read.Close();
            fileRead.Close();
            fileSave.Close();
            File.Delete(@"Books.dat");
            File.Move(@"BooksTMP.dat", @"Books.dat");
        }
        catch{}
    }
    var Refresh = this.Owner as MainWindow;
    Refresh.Refresh();
    this.Close();
}
```

W ten sposób możemy dodać bardzo dużą liczbę książek, ponieważ są one przechowywane w bazie danych "Books.dat"

Baza studentów została wykonana w ten sam sposób:


Biblioteka
Daj ksiazke
Dodac ucznia do bazy
Student zwr
Usun ucznia z bazy
Dodac Studenta
Name:
Artem
Lastname:
Levchenko
Kierunek:
WEAII
Nr. albumu:
97923
CANCEL
ACCEPT
Black List


Biblioteka
Daj ksiazke uczniowi
Dodac ksiazke do bazy
Dodac ucznia do bazy
Student zwrocil ksiazke
Usun ksiazke z bazy
Usun ucznia z bazy
Artem Levchenko WEAil 97923
Black List


Biblioteka
Dodac ksiazke do bazy
Dodac ucznia do bazy
Usun ksiazke z bazy
Usun ucznia z bazy
Usun studenta
Name:
Artem
Lastname:
Levchenko
Nr. albumu:
97923
CANCEL
ACCEPT
Artem Levchenko WEAil 97923
Black List

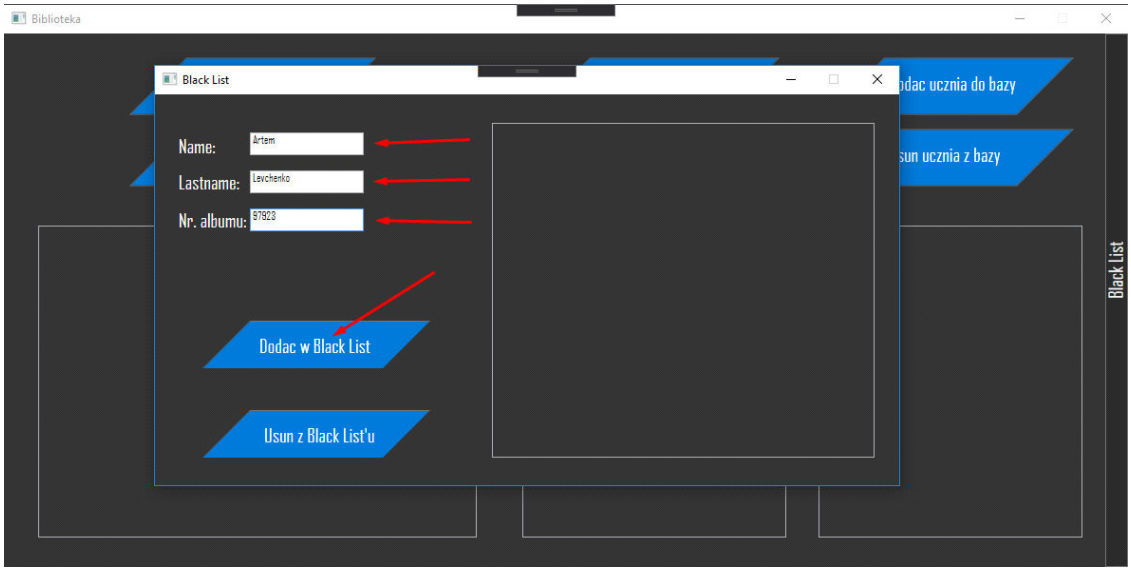
"Dodac ucznia do bazy":

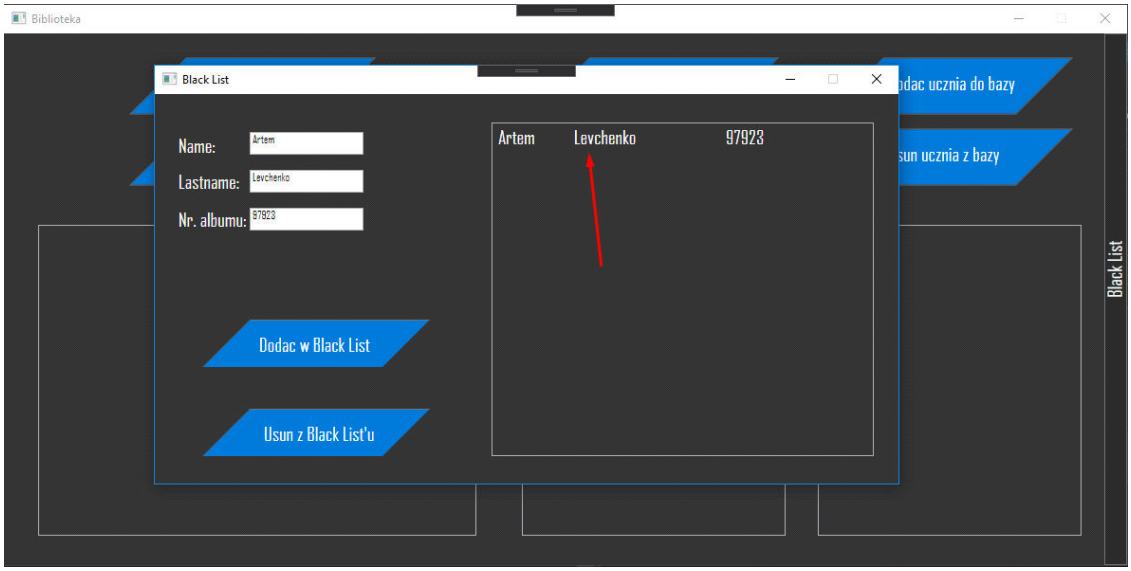
```
ссылка: 1
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (name.Text != String.Empty & lastname.Text != String.Empty & kierunek.Text != string.Empty & NRalbumu.Text != string.Empty)
    {
        var BL = this.Owner as MainWindow;
        if (BL.BL(name.Text, lastname.Text, NRalbumu.Text) != false)
        {
            Students student = new Students(name.Text, lastname.Text, kierunek.Text, NRalbumu.Text);
            BinaryFormatter formatter = new BinaryFormatter();
            FileStream file = new FileStream(@"Students.dat", FileMode.Append);
            formatter.Serialize(file, student);
            file.Close();
            this.Close();
        }
        else
        {
            MessageBox.Show("Error, student in BlackList!");
        }

    }
    var Refresh = this.Owner as MainWindow;
    Refresh.Refresh();
}
```

"Usun ucznia z bazy":

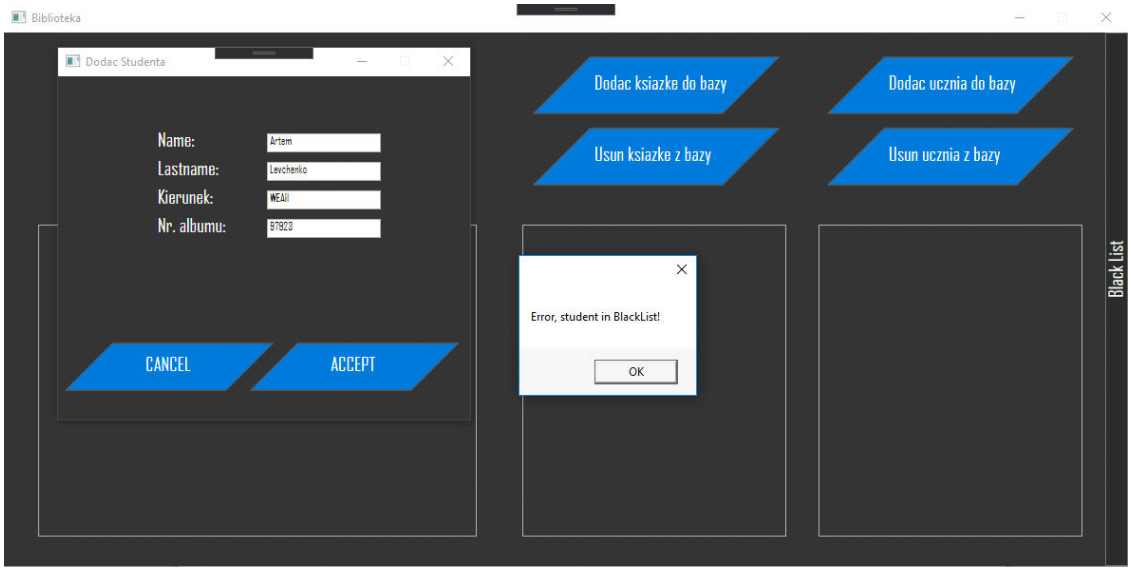
```
ссылка: 1
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (name.Text != String.Empty & lastname.Text != String.Empty & NRalbumu.Text != String.Empty)
    {
        BinaryFormatter formatter = new BinaryFormatter();
        FileStream fileRead = new FileStream(@"Students.dat", FileMode.OpenOrCreate);
        FileStream fileSave = new FileStream(@"StudentsTMP.dat", FileMode.OpenOrCreate);
        BinaryReader read = new BinaryReader(fileRead);
        try
        {
            while (read.PeekChar() >= 0)
            {
                Students student = (Students)formatter.Deserialize(fileRead);
                if (name.Text == student.Name & lastname.Text == student.LastName & NRalbumu.Text == student.NRalbumu) { }
                else
                {
                    formatter.Serialize(fileSave, student);
                }

            }
            read.Close();
            fileRead.Close();
            fileSave.Close();
            File.Delete(@"Students.dat");
            File.Move(@"StudentsTMP.dat", @"Students.dat");
        }
        catch { }
    }
    var Refresh = this.Owner as MainWindow;
    Refresh.Refresh();
    this.Close();
}
```

Zaimplementowano również czarną listę:





Gdy uczeń jest na czarnej liście, nie możemy dodać go do bazy danych:



Realizacja dodania ucznia do czarnej listy:

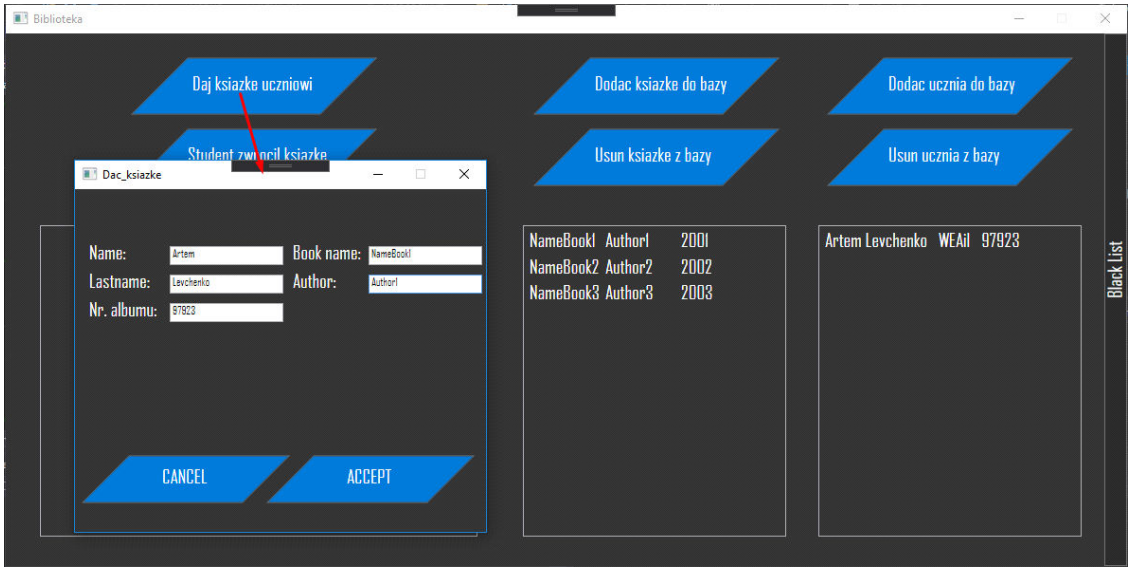
```
ссылка: 1
private void Button_Click(object sender, RoutedEventArgs e)
{
    if (Name.Text != String.Empty & Lastname.Text != String.Empty & NRalbumu.Text != String.Empty)
    {
        Students student = new Students(Name.Text, Lastname.Text, null, NRalbumu.Text);
        BinaryFormatter formatter = new BinaryFormatter();
        FileStream file = new FileStream(@"BL.dat", FileMode.Append);
        formatter.Serialize(file, student);
        file.Close();
        Refresh();
    }
}
```

Realizacja usunięcia ucznia z czarnej listy:

```
ссылка: 1
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (Name.Text != String.Empty & Lastname.Text != String.Empty & NRalbumu.Text != String.Empty)
    {
        BinaryFormatter formatter = new BinaryFormatter();
        FileStream fileRead = new FileStream(@"BL.dat", FileMode.OpenOrCreate);
        FileStream fileSave = new FileStream(@"BLtmp.dat", FileMode.OpenOrCreate);
        BinaryReader read = new BinaryReader(fileRead);
        try
        {
            while (read.PeekChar() >= 0)
            {
                Students student = (Students)formatter.Deserialize(fileRead);
                if (Name.Text == student.Name & Lastname.Text == student.LastName & NRalbumu.Text == student.NRalbumu) { }
                else
                {
                    formatter.Serialize(fileSave, student);
                }

            }
            read.Close();
            fileRead.Close();
            fileSave.Close();
            File.Delete(@"BL.dat");
            File.Move(@"BLtmp.dat", @"BL.dat");
        }
        catch { }
    }
    Refresh();
}
```

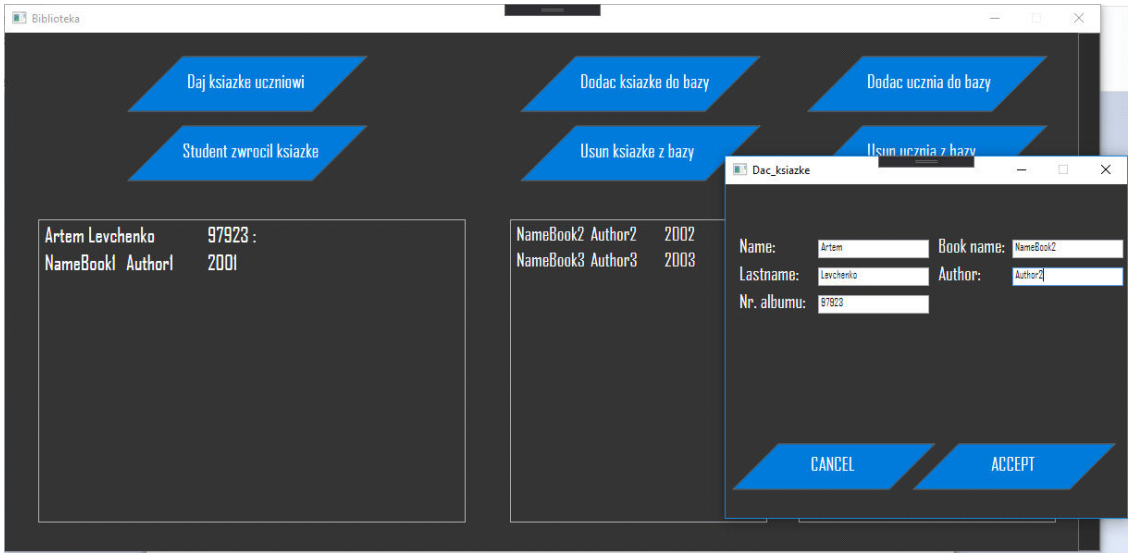
Na przykład mamy 3 książki i 1 ucznia(W moim programie uczeń może wziąć maks. 3 książki):

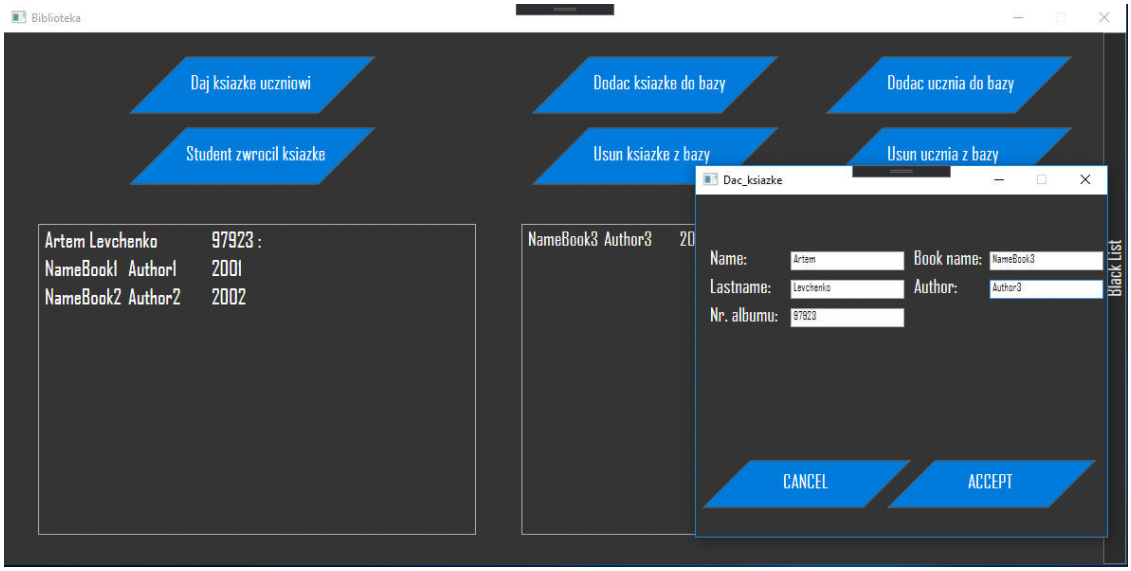
Klikając na "Dak ksiazke uczniowi" mamy możliwość przekazania książek dla studentów:



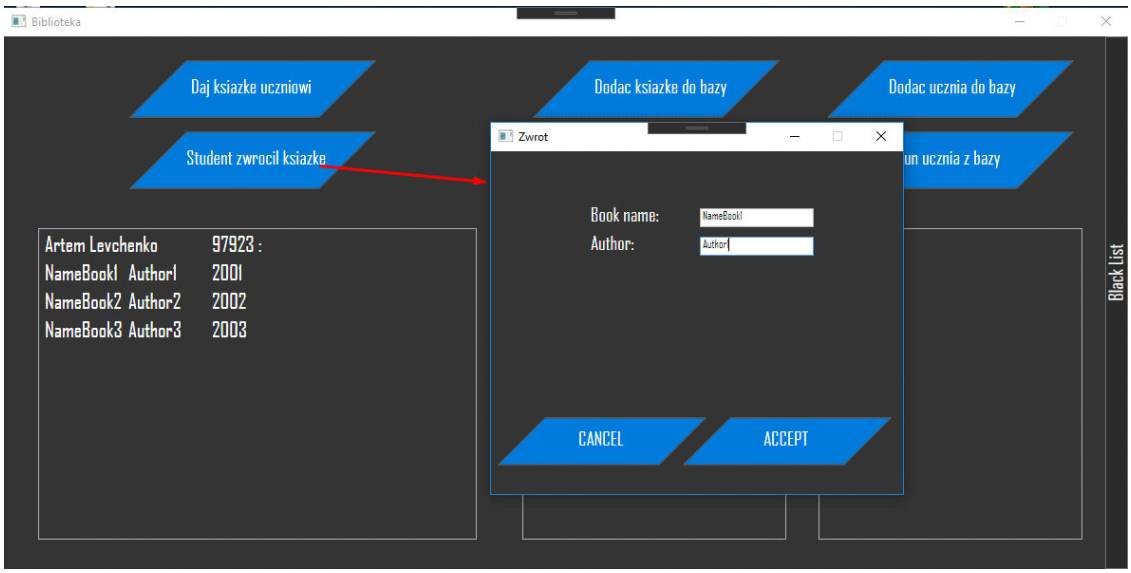
Klikając "ACCEPT" mamy taki wynik:

Na przykład student chciał wziąć jeszcze 2 książki:



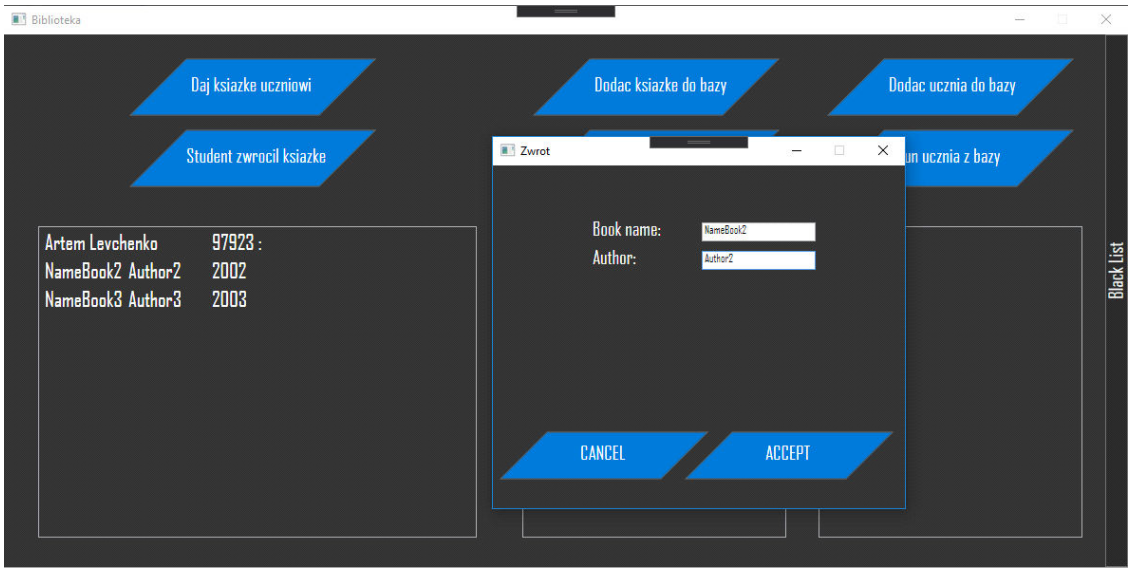


A potem student chciał zwrócić jedną książkę, na przykład "NameBook1":

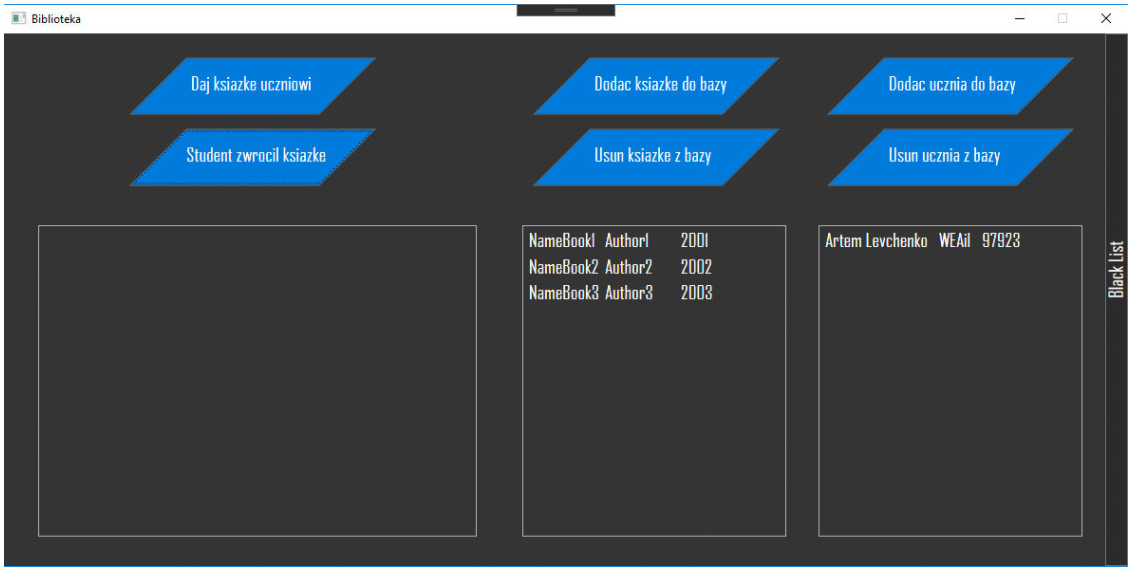


Książka wraca do biblioteki:

A potem zwraca 2 kolejne książki:



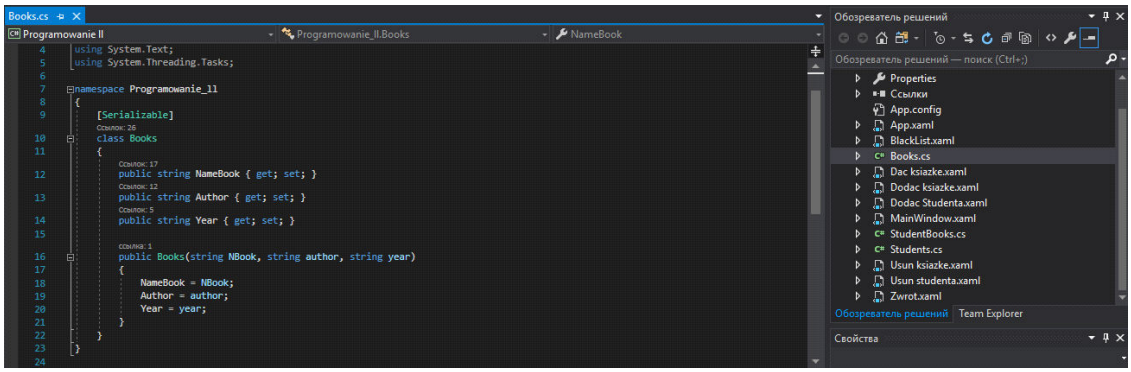
(A kiedy uczeń zwróci ostatnią pobraną książkę, zostanie automatycznie dodany do bazy danych studentów)

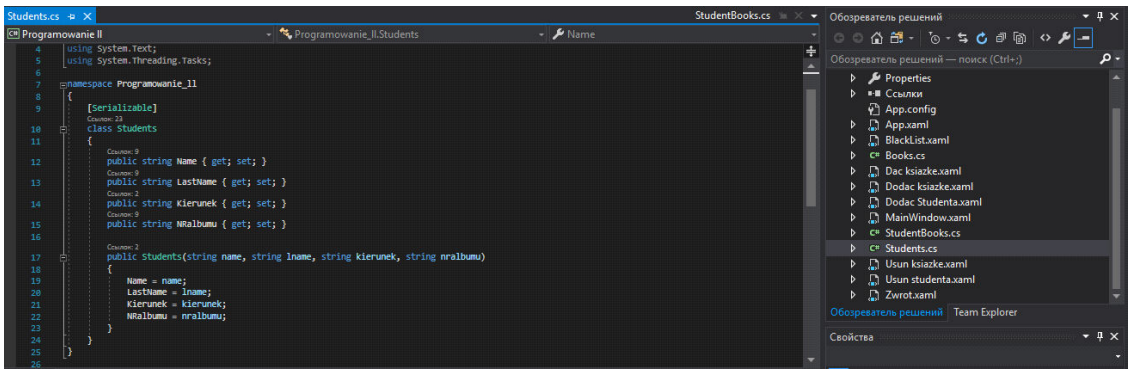


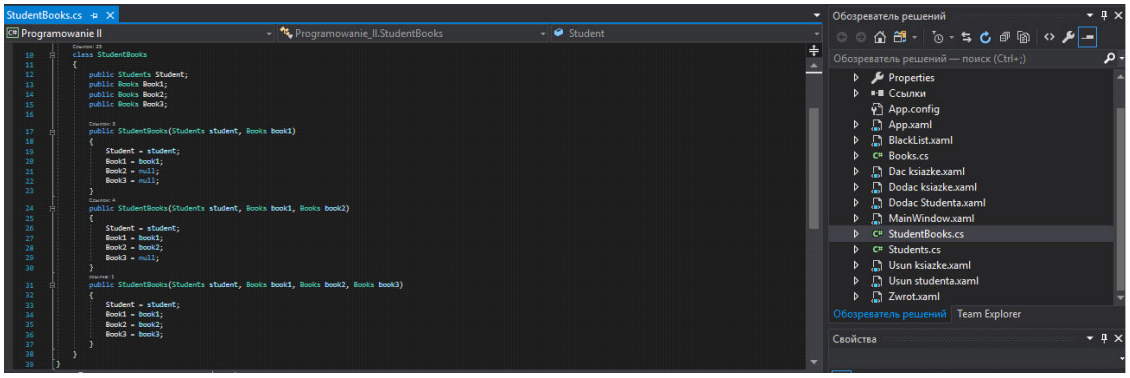
Kod do wydawania i zwracania książek jest bardzo duży do zdjęć lub wklejenia tutaj, więc wyślę Panu mój projekt.

Wniosek:

Główne funkcje aplikacji są realizowane za pomocą 3 klasow: "Books", "Students", "StudentBooks".







Aby zaktualizować i wyświetlić dane z bazy danych, korzystam z funkcji Refresh ();

Refresh() dla BlackList:

```
Ссылок: 3
public void Refresh()
{
    Blist.Items.Clear();
    BinaryFormatter formatter = new BinaryFormatter();
    FileStream file = new FileStream(@"BL.dat", FileMode.OpenOrCreate);
    BinaryReader read = new BinaryReader(file);
    string tmp;
    try
    {
        while (read.PeekChar() >= 0)
        {
            Students student = (Students)formatter.Deserialize(file);
            tmp = student.Name + "\t";
            tmp += student.LastName + "\t\t";
            tmp += student.NRalbumu;
            Blist.Items.Add(tmp);
        }
    }
    catch
    {
    }
    read.Close();
    file.Close();
}
```

Refresh() dla MainWindow:

```
Ссылок: 7
public void Refresh()
{
    List2.Items.Clear();
    BinaryFormatter formatter = new BinaryFormatter();
    FileStream file = new FileStream(@"Books.dat", FileMode.OpenOrCreate);
    BinaryReader read = new BinaryReader(file);
    string tmp;
    try
    {
        while(read.PeekChar() >= 0)
        {
            Books book = (Books)formatter.Deserialize(file);
            tmp = book.NameBook + "\t";
            tmp += book.Author + "\t";
            tmp += book.Year;
            List2.Items.Add(tmp);
        }
    }
    catch
    {
    }
    read.Close();
    file.Close();
```

```
List1.Items.Clear();
FileStream file1 = new FileStream(@"SB.dat", FileMode.OpenOrCreate);
BinaryReader read1 = new BinaryReader(file1);
try
{
    while (read1.PeekChar() >= 0)
    {
        StudentBooks sb = (StudentBooks)formatter.Deserialize(file1);
        tmp = sb.Student.Name + " ";
        tmp += sb.Student.LastName + "\t";
        tmp += sb.Student.NRalbumu + " :";
        List1.Items.Add(tmp);
        tmp = sb.Book1.NameBook + "\t";
        tmp += sb.Book1.Author + "\t";
        tmp += sb.Book1.Year;
        List1.Items.Add(tmp);
        if(sb.Book2 != null)
        {
            tmp = sb.Book2.NameBook + "\t";
            tmp += sb.Book2.Author + "\t";
            tmp += sb.Book2.Year;
            List1.Items.Add(tmp);
        }
        if(sb.Book3 != null)
        {
            tmp = sb.Book3.NameBook + "\t";
            tmp += sb.Book3.Author + "\t";
            tmp += sb.Book3.Year;
            List1.Items.Add(tmp);
        }
    }
}
catch
{
}
read1.Close();
file1.Close();
```

```
    List3.Items.Clear();
    FileStream file3 = new FileStream(@"Students.dat", FileMode.OpenOrCreate);
    BinaryReader read3 = new BinaryReader(file3);
    try
    {
        while (read3.PeekChar() >= 0)
        {
            Students student = (Students)formatter.Deserialize(file3);
            tmp = student.Name + " ";
            tmp += student.LastName + "    ";
            tmp += student.Kierunek + "    ";
            tmp += student.NRalbumu;
            List3.Items.Add(tmp);
        }
    }
    catch {}
    read3.Close();
    file3.Close();
}
```