**Съдържание**

**1. Въведение**

**2. Използвани технологии – програмни езици, библиотеки /кратко описание на Entity Framewok, какви възможности предоставя; Windows Forms/**

**3. Описание на приложението**

**3.1. Кратко описание**

**3.2. Основни функции – описание на възможностите на вашата програма**

**3.3. Основни модули и връзки между тях**

**4.    Архитектура на проекта. Реализация.**

**4.1. Бази данни – описание на използвани класове /entity/, таблици, име на базата данни, подход /CodeFirst или DatabaseFirst/, dbContext, снимка на структурата на БД, с връзките между таблиците, вид на връзките**

**4.2. Потребителски интерфейс – описание на всички windows форми, които сте създали, снимки на формите /до 3 бр./**

**4.3. Бизнес слой  - описание на класовете и техните методи за реализиране на CRUD операциите.**

**5.   Заключение /бъдещо развитие на проекта/**

**1. Въведение**

Целта на формичката е да помага на хората да намарят книгата която те искат да четат и да го правят онлайн. Има собствена потребителска система, но тя още няма полза понеже формичката не е завършена и няма време да я завърша, общо взето потребителската система е безполезна, ако не се полза за заемането на книга, което не е направено кой знае колко добре, но си върши работата.



**2. Използвани технологии – програмни езици, библиотеки /кратко описание на Entity Framewok, какви възможности предоставя;**

**Windows Forms/**

**Какво е Entity Framework?**

Entity Framework е набор от технологии в ADO.NET, които поддържат разработването на софтуерни приложения, ориентирани към данни. Архитектите и

разработчиците на приложения, ориентирани към данни, се борят с необходимостта да постигнат две много различни цели. Те трябва да моделират обектите, връзките и логиката на бизнес проблемите, които решават, и трябва също да работят с машините за данни, използвани за съхраняване и извличане на данните. Данните могат да обхващат множество системи за съхранение, всяка със собствени протоколи; дори приложения, които работят с една система за съхранение, трябва да балансират изискванията на системата за съхранение спрямо изискванията за писане на ефективен и поддържаем код на приложение.

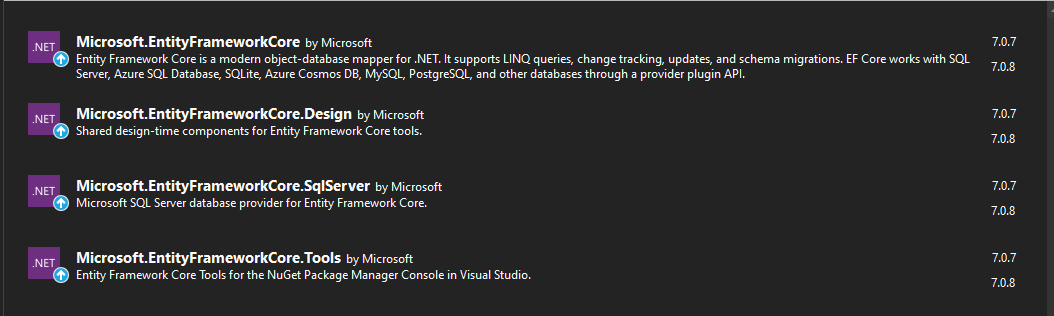
**Какво е Windows Forms?**

Windows Forms (WinForms) е безплатна библиотека с графични (GUI) класове с отворен код, включена като част от Microsoft .NET, .NET Framework или Mono, предоставяща платформа за писане на клиентски приложения за настолен компютър, лаптоп и таблет Компютри. Въпреки че се разглежда като заместител на по-ранната и по-сложна C++ базирана Microsoft Foundation Class Library, тя не предлага сравнима парадигма и действа само като платформа за нивото на потребителския интерфейс в многослойно решение.

**Използвани технологии:**

**Ползвах езика C#.Библиотека- Windows Forms .Net version 6.0**

**Ползвах също и тези NuGet Packages**

**Работя по модел Code-First където специалното е просто, че създаваме базата данни чрез програмен код.**

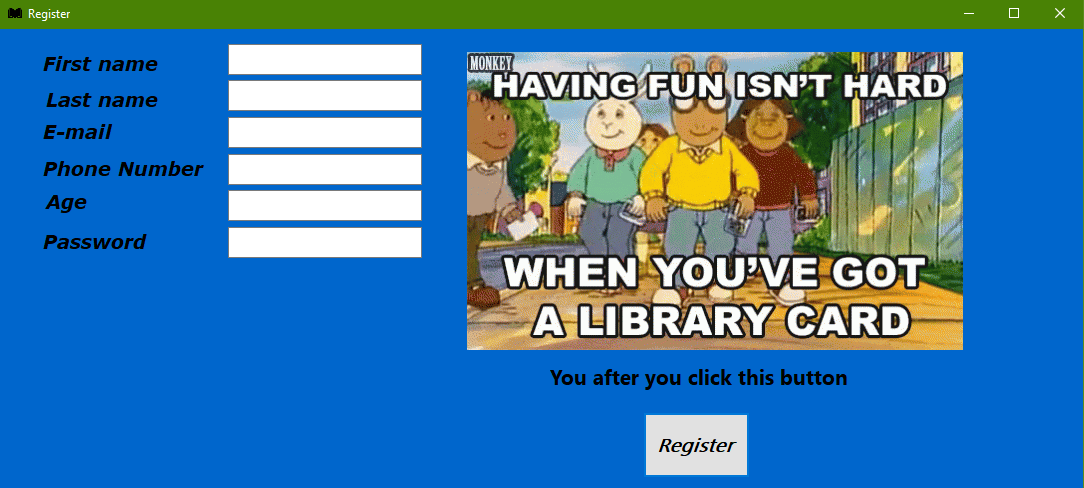
**3.Описание на приложението**

В главното меню има 5 бутона, които имат различни функции сега ще обясня как работи всяка една от тях и за какво служи:

3.1/Register(Регистрация)

****

Когато натиснете този бутон то ще ви пренасочи в друго меню и там ще видите следното:



Вече сте в менюто за регистрация тук можете да си направите членска карта за да заемате книгите от другото меню по късно в тази документация. Всичко което трябва да направите е да си дадете данните в текстовите кутии.

3.2/Register a book(Регистрация на книга)



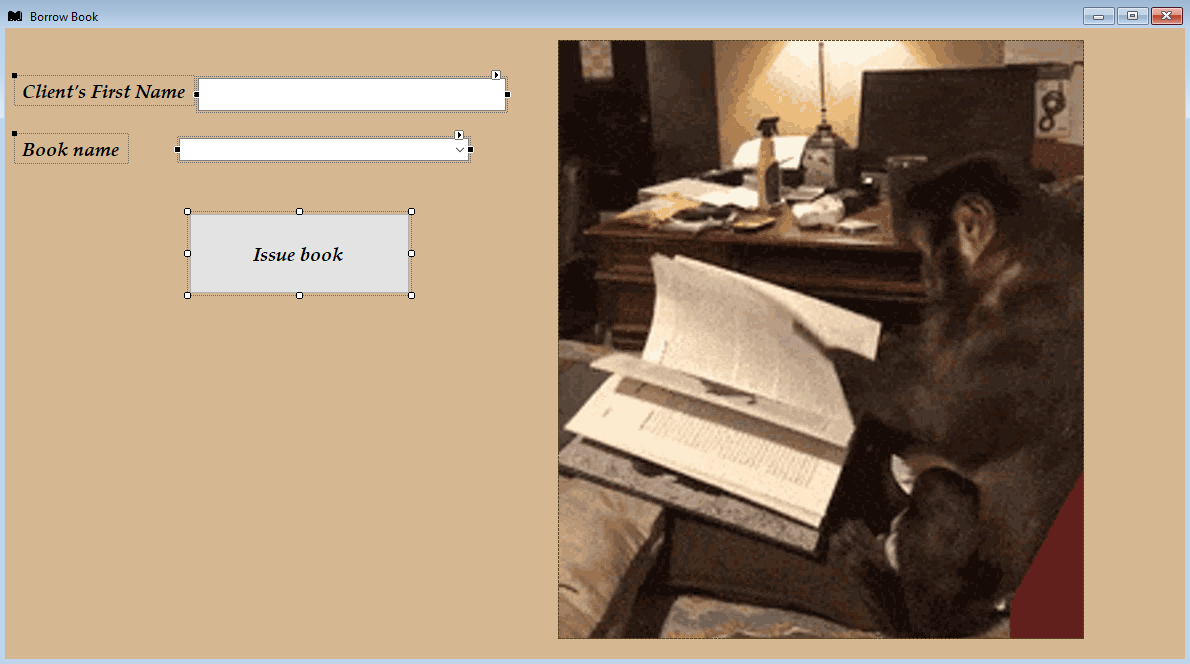
Като натиснете този бутон то ще ви пренасочи в друго меню и там ще видите следното:



Вече сте си създал/а членска карта, защо не и да допринесете на като дарите(създадете) книга за библиотеката и това става много лесно просто попълнете информацията за съответната книга и натиснете бутона за регистрация на книга! Толкова лесно.

3.3/Заемане на книга (Borrow book)

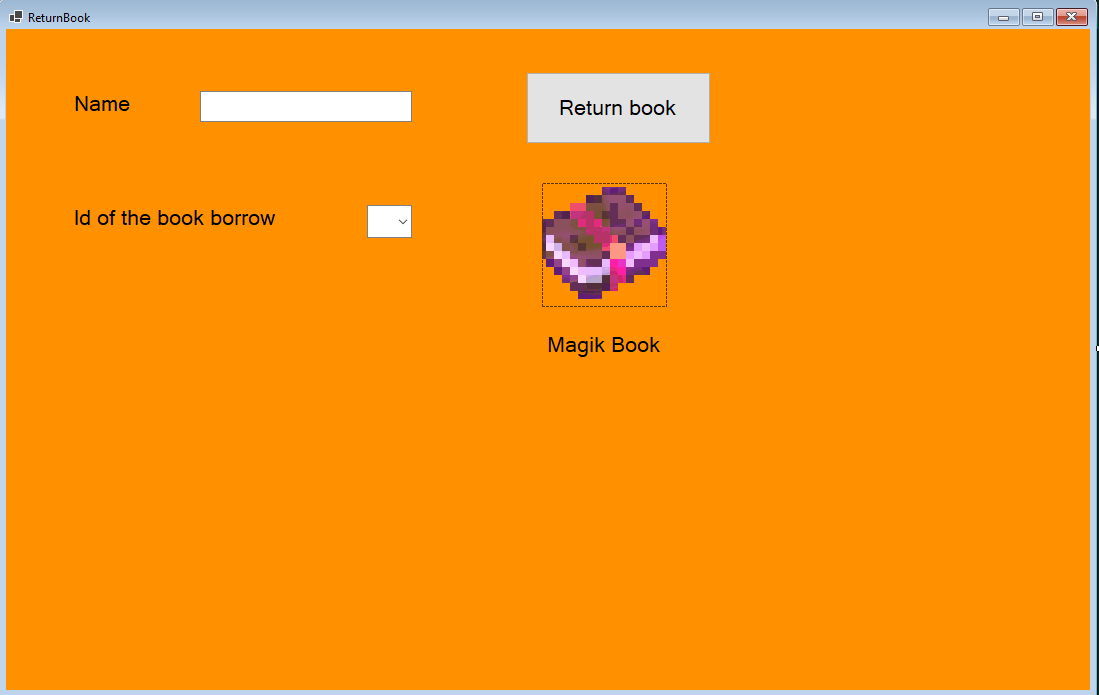
Като натиснете този бутон то ще ви пренасочи в друго меню и там ще видите следното:



Под разбираемо е, че една библиотека не е нищо ако не можеш да вземаш книги от нея и тази формичка прави точно това попълнете името си и изберете въпросната книга и натиснете бутона Заемане. Лесно нали?

3.4/ Връщане на книга(Return book)//В процес



Като натиснете този бутон то ще ви пренасочи в друго меню и там ще видите следното:

Тук нещата ще ги държа прости даваш си името и номера на заемане на книгата ви и натиснете бутона за завръщане но все още тази не е завършена само от към Back-end code

3.5 Turn off(Изключване)

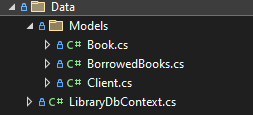


този бутон не визуализира нищо просто ви изключва формичката ╮ (. ❛ ᴗ ❛.) ╭

**4.    Архитектура на проекта. Реализация.**

**4.1/ Бази данни**

Ентититата, които използвах са следните:

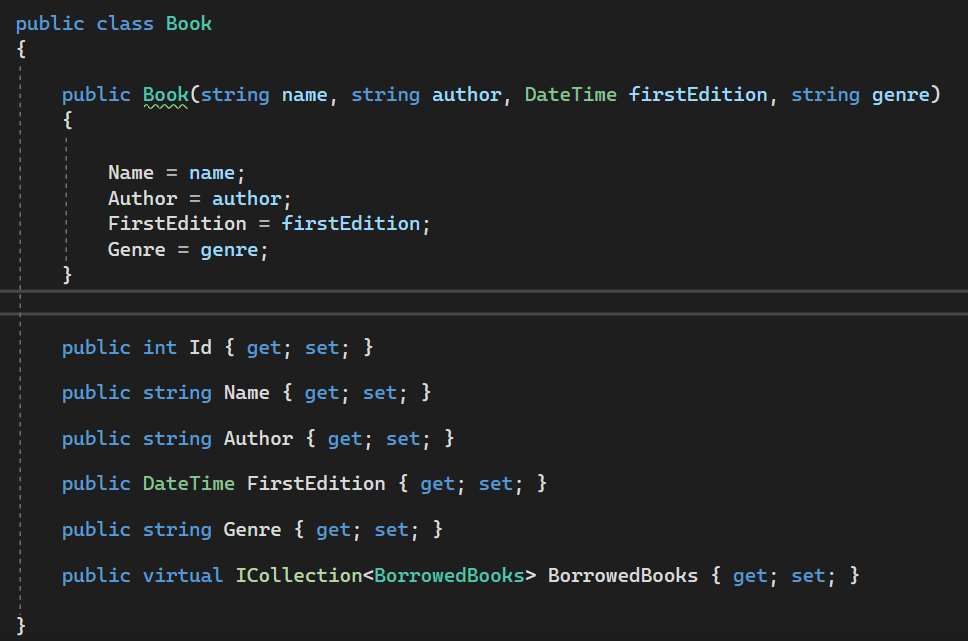
****

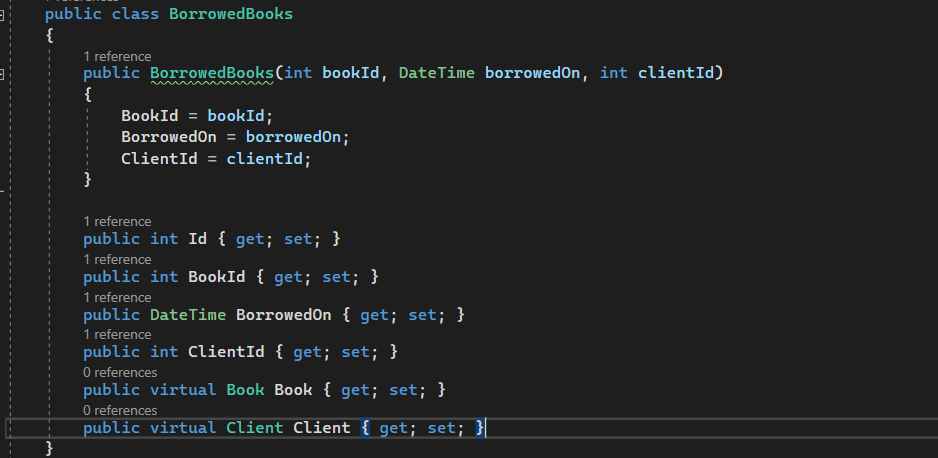
Има връзка между:

Book и BorrowBook – OneToMany;

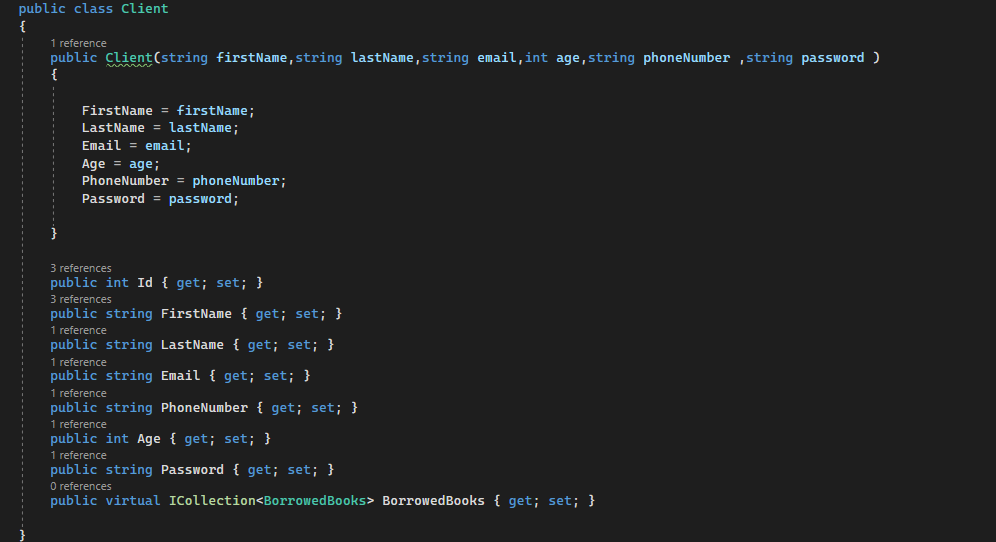
Client и BorrowBook – OneToMany.

Класът Book.cs

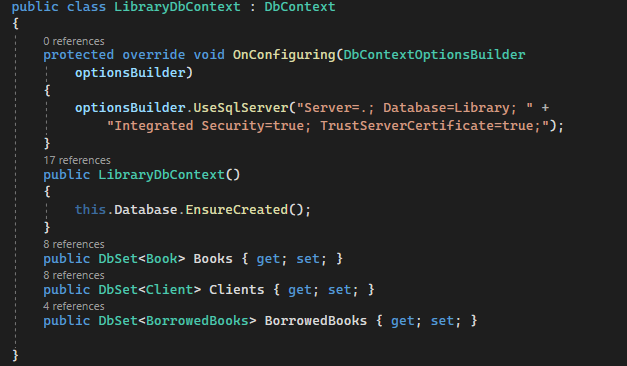
Класът BorrowBook.cs

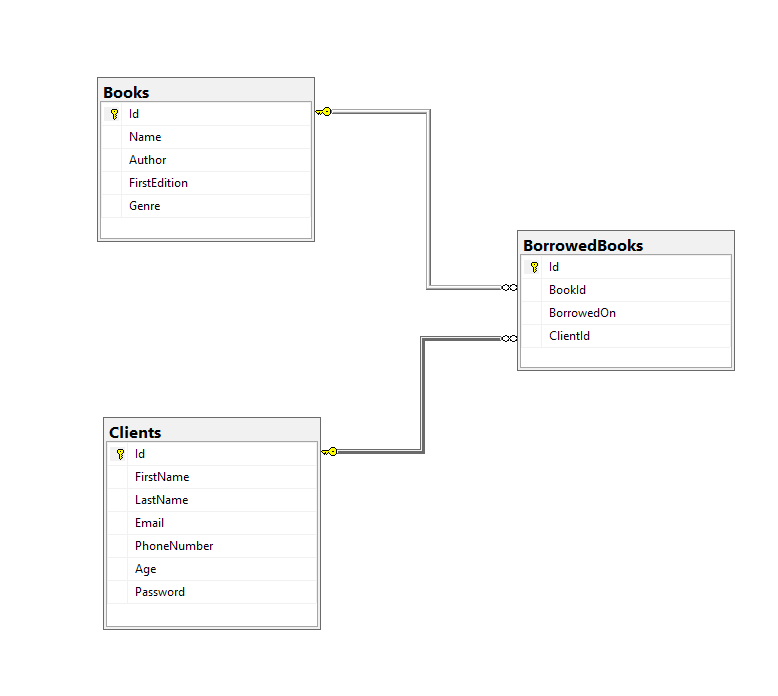


Класът Client.cs



LibraryDbContext.cs



Ето и малка диаграма на как точно изглеждат връзките в базата данни:

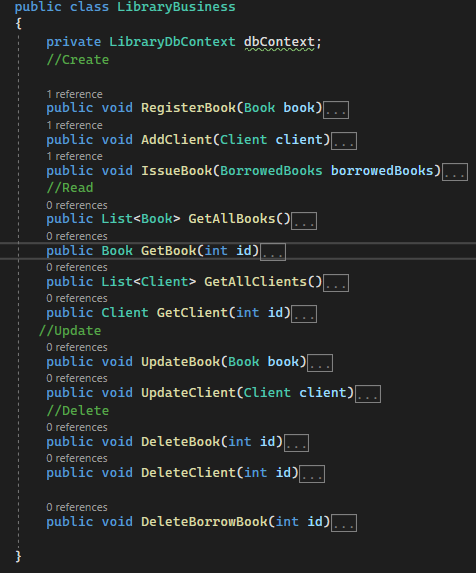
И това е всичко което мога да покажа за базата.

4.2/  **Бизнес слой**

Това е моят бизнес слой в неговата папка:



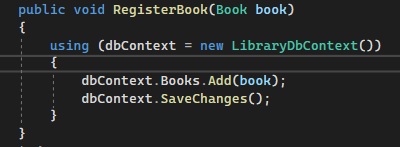
Това са всичките ми методи главно са CRUD операции:



Сега ще мина през всеки един и ще кажа какво прави на кратко:

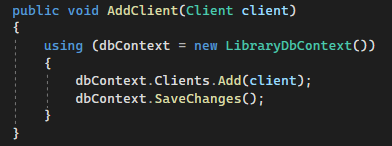
Create методи

**1.RegisterBook(Book book);**



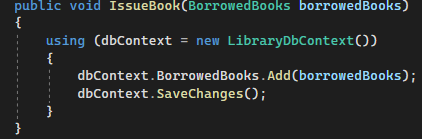
Достъпваме базата и създаваме обект от ентитито Book.

**2.AddClient(Client client)**

****

Достъпваме базата и създаваме обект от ентитито Client.

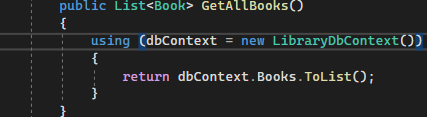
3.IssueBook(BorrowedBooks borrowedBooks)

****

Достъпваме базата и създаваме обект от ентитито BorrowedBook.

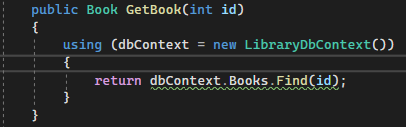
Read методи

**1.**GetAllBooks()



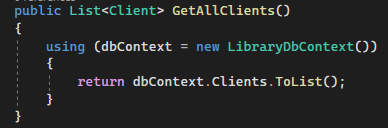
Това просто прави лист от всички книги.

**2.GetBook(int id)**



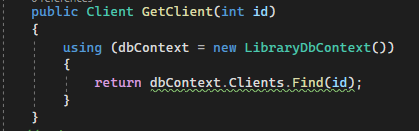
Това просто дава възможноста да търсим книга с Id.

**3. GetAllClients()**

****

Това просто прави лист от всички клиенти.

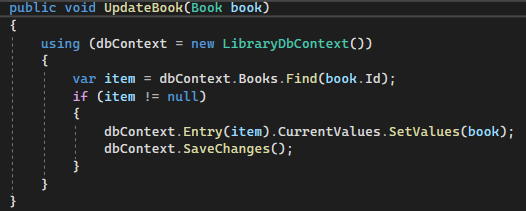
**4.GetClient(int id)**

****

Това просто дава възможноста да търсим книга с Id

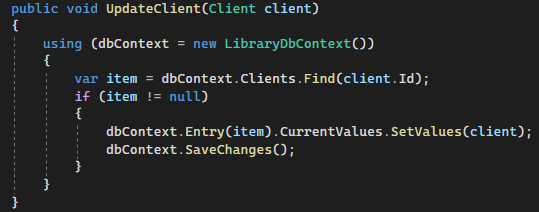
Update методи

**1.UpdateBook(Book book)**

****

С това достъпваме базата и този метод ни дава възможноста да променим обект от ентити Book.

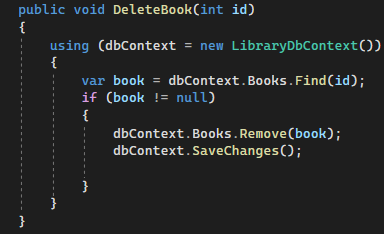
**2.UpdateClient(Client client)**

****

С това достъпваме базата и този метод ни дава възможноста да променим обект от ентити Client.

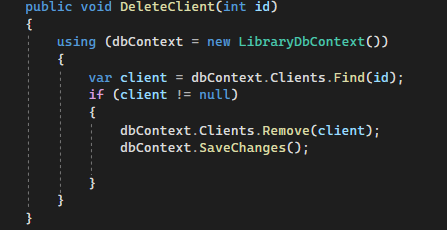
Delete методи

**1.DeleteBook(int id)**

****

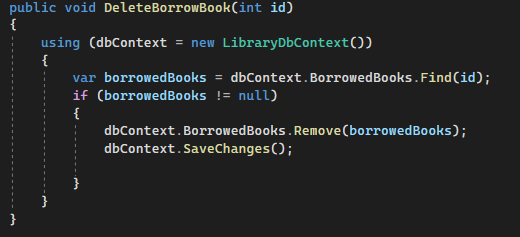
**Тази функция търси Id от обект от ентити Book, ако Id-то съответства на съществуващ обект то обекта се изтрива.**

**2.DeleteClient(int id)**

****

**Тази функция търси Id от обект от ентити Book, ако Id-то съответства на съществуващ обект то обекта се изтрива.**

**3.DeleteBorrowBook(int id)**

****

**Тази функция търси Id от обект от ентити BorrowedBook, ако Id-то съответства на съществуващ обект то обекта се изтрива.**

**5. В заключение има какво да се подобри в формичката, но мързелът ми е в повече**