Szkoła Główna Gospodarstwa Wiejskiego

w Warszawie

Wydział Zastosowań Informatyki i Matematyki

Sebastian Łyszkowski

Numer albumu: 191698

Aplikacja wspierająca korzystanie z biblioteki oraz wspomagająca jej pracę

Application supporting the use of the library

and supporting her work

Praca dyplomowa inżynierska

na kierunku – Informatyka

Praca wykonana pod kierunkiem

dr. Andrzeja Zembrzuskiego

Wydział Zastosowań Informatyki i Matematyki

Katedra Systemów Informatycznych

Zakład Symulacji Komputerowych

Warszawa, 2021 rok

**Oświadczenie promotora pracy**

Oświadczam, że niniejsza pracazostała przygotowana pod moim kierunkiem i stwierdzam, że spełnia warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data .................................... Podpis promotora pracy ...................................................

**Oświadczenie autora pracy**

Świadom odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowazostałanapisanaprzeze mnie samodzielnie i nie zawieratreści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. Nr 90 poz. 631 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną. Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiatowej.

Data ..................................... Podpis autora pracy .............................................

**Streszczenie**

**Aplikacja wspierająca korzystanie z biblioteki oraz wspomagająca jej pracę**

Celem niniejszej pracy było wytworzenie aplikacji umożliwiającej pracownikom biblioteki zdalne zarządzanie nią, a czytelnikom zdalne korzystanie z niej. Aplikacja została napisana w języku C# z wykorzystaniem platformy .NET. Przy tworzeniu baz danych do aplikacji zostało zastosowane podejście „Code First”. Praca zaczyna się od wstępu teoretycznego poświęconego historii i działaniu technologii wykorzystanych przy tworzeniu aplikacji, takich jak: C#, .NET, Entity Framework. Główną częścią pracy jest omówienie funkcjonalności programu oraz ich implementacji. Na końcu pracy znalazła się ocena jakości wytworzonej aplikacji, w której zostały przedstawione zalety i wady stworzonej aplikacji oraz omówienie perspektyw jej rozwoju.

Słowa kluczowe – C#, .NET, Entity Framework, Windows Forms, T-SQL.

**Summary**

**Application supporting the use of the library and supporting her work**

The aim of this work was to create an application that would allow library employees to remotely manage it, and readers to use it remotely. The application was written in C# using the .NET platform. The "Code First" approach was applied to the application when creating the databases. The work begins with a theoretical introduction devoted to the history and usage of technologies used in the development of applications, such as: C#, .NET, Entity Framework. The main part of the work is to discuss the features of the program and their implementation. At the end of the work, the quality of the created application was assessed, in which the advantages and disadvantages of the created application were presented, as well as the prospects for its development.

Keywords – C#, .NET, Entity Framework, Windows Forms, T-SQL.

Spis treści

[1. Wstęp 8](#_Toc57196041)

[1.1. Cel i ogólny opis projektu 8](#_Toc57196042)

[1.2. Istniejące rozwiązania na rynku 9](#_Toc57196045)

[1.3. Porównanie aplikacji 9](#_Toc57196049)

[1.4. Omówienie technologii wykorzystanych w projekcie 9](#_Toc57196050)

[1.5. Omówienie struktury aplikacji 12](#_Toc57196051)

[1.5.1. Omówienie struktury bazy danych 13](#_Toc57196058)

[1.5.2. Omówienie logiki aplikacji 15](#_Toc57196059)

[2. Bibliografia 16](#_Toc57196060)

# Wstęp

## Cel i ogólny opis projektu

Celem niniejszej pracy jest zaprojektowanie oraz stworzenie aplikacji, która wspierałaby pracę bibliotekarzy w zarządzaniu biblioteką i ułatwiałaby korzystanie z niej czytelnikom. Aplikacja jest przeznaczona na najpopularniejszy system operacyjny dla komputerów stacjonarnych – Windows 10. Aplikacja musi posiadać przejrzysty oraz intuicyjny interfejs, aby ludzie młodzi, jak i starsi nie mieli problemu z korzystaniem z aplikacji.

Aplikacja musi posiadać dwa typy kont: konto administratora (bibliotekarza) i konto zwykłego użytkownika (czytelnika). Konta posiadają różne uprawnienia, konto administratora posiada wszystkie funkcje konta użytkownika oraz specjalne dodatkowe funkcje administratora.

Zwykły użytkownik posiada następujące funkcje:

* przeglądanie bazy książek,
* rezerwowanie książek,
* przeglądanie historii swoich wypożyczeń,
* przeglądanie płatności.

Administrator oprócz funkcji zwykłego użytkownika posiada też:

* edytowanie informacji o bibliotece,
* zmienianie ustawień działania biblioteki,
* edytowanie tabeli książek i sygnatur,
* przeglądanie bazy danych,
* nakładanie oraz zdejmowanie blokad z kont użytkowników,
* zatwierdzanie płatności,
* zatwierdzanie wypożyczeń i zwrotów.

Aplikacja jest połączona z dwiema bazami danych: bazą kont wszystkich użytkowników oraz bazą wszystkich książek biblioteki. Jako język bazodanowy wybrałem T-SQL używany w platformie Microsoft SQL Server, z którego korzystałem do tworzenia baz danych.

## Istniejące rozwiązania na rynku

Obecnie na rynku istnieje wiele różnorodnych rozwiązań o tej samej tematyce. Warto przed tworzeniem własnej aplikacji zwrócić uwagę na te rozwiązania i upewnić się, że czy zaproponowane rozwiązanie już istnieje na rynku. Warto też sprawdzić jakie funkcjonalności powinna implementować dobra aplikacja, aby odpowiadała na żądania użytkowników. Dlatego warto sprawdzić co oferuje rynek i porównać to z naszym rozwiązaniem.

### Bibliotekarz.NET



Rys. 1. Wygląd aplikacji Bibliotekarz.NET.

Źródło: Strona internetowa autora[[1]](#_Bibliografia).

Bibliotekarz.NET to platforma rozwiązań bibliotecznych autorstwa Marka Ryńskiego. Powstała w 2004 roku, a najnowsza wersja (1.6.0.7) z 2014 roku. Program posiada intuicyjny interfejs (Rys. 1) oraz charakteryzuje się wysoką wydajnością i bezpieczeństwem. Aplikacja jest całkowicie darmowa. System wspiera pracę bibliotek szkolnych, uczelnianych, firmowych, kościelnych, publicznych oraz własnych niedużych księgozbiorów. Twórca uważa, że system cechują następujące zalety:

* “dodawanie, usuwanie i edycja czytelników oraz publikacji,
* rozbudowane wyszukiwanie czytelników oraz publikacji,
* wypożyczanie, zwracanie, prolongowanie publikacji,
* możliwość korzystania z kilku baz danych,
* określanie lokalizacji, stanu oraz tematyki publikacji,
* statystyki graficzne,
* wycofywanie (ubytkowanie) publikacji,
* raporty PDF, XLS, HTML: zaległości, wypożyczeń, itp.,
* obsługa kodów kreskowych,
* pobieranie opisów książek na podstawie numerów ISBN z Internetu,
* wydruki: bazy publikacji, listy zaległości, listy czytelników i inne,
* zabezpieczenie programu przed osobami nieupoważnionymi,
* archiwizacja kopii bezpieczeństwa bazy,
* obsługa skontrum,
* powiadamianie czytelników via e-mail o zaległościach.”[[1]](#_Bibliografia)

### Prolib

Prolib to system informatyczny stworzony w przez informatyków firmy Max Elektronik S.A. (aktualnie Sygnity Business Solutions). Aplikacja jest przeznaczona do obsługi i zarządzania różnego rodzaju bibliotek. Prolib jest komercyjnym rozwiązaniem wciąż rozwijanym przez dużą firmę i jest używany w wielu placówkach (na czas pisania pracy jest to 285). Firma gwarantuje polskojęzyczny system wsparcia i usługi serwisowe. System posiada takie funkcjonalności jak:

* “Szybka i wieloaspektowa obsługa czytelników.
* Pozyskiwanie, opracowywanie, wyszukiwanie i ocena informacji.
* Wsparcie dla administracyjnych i informatycznych obszarów działalności biblioteki.
* Gromadzenie, opracowywanie, udostępnianie tradycyjnych i cyfrowych zasobów bibliotek.
* Optymalizacja pracy innych działów instytucji. Integracja systemu bibliotecznego z innymi systemami, np. działu kadr, księgowości, dziekanatu i działu IT.
* Umożliwienie użytkownikom samodzielnego i efektywnego zarządzania eksploatacją systemu bibliotecznego, jego skalowalności i adaptacji do aktualnej sytuacji biblioteki w odpowiedzi na rozwój i zmiany organizacyjne instytucji.”[[2]](#_Bibliografia)



## Porównanie aplikacji

## Omówienie technologii wykorzystanych w projekcie

W tworzeniu aplikacji w głównej mierze został wykorzystany język C#. Jest to wysokopoziomowy, zorientowany obiektowo język programowania ogólnego przeznaczenia stworzony przez firmę Microsoft w roku 2000 i jest nadal rozwijany. Najnowszą wersją języka C# na czas pisania pracy jest C# 9.0 (premiera w październiku 2020 roku) i właśnie ta wersja jest wykorzystywana w aplikacji. Język C# jest bardzo podobny do języka Java o czym świadczy stwierdzenie „*Java and C# are almost identical programming languages.*”[[3]](#_Bibliografia). Głównymi cechami języka C# są:

* Obiektowość z dziedziczeniem – przekazywanie cech klasy bazowej do klasy podrzędnej. Tak jak w języku Java, dziedziczenie opiera się na jednej klasie nadrzędnej, po której dziedziczą inne klasy. W języku C# jest to System.Object.
* Garbage Collection – system zarządzania pamięcią, usuwa referencje do nieużywanych obiektów, dzięki czemu w większości przypadków programista nie musi się troszczyć o zwalnianie pamięci poprzez wywoływanie destruktorów.
* Typy ogólne (ang. *Generics*) – Typy ogólne umożliwiają dostosowanie metody, klasy, struktury lub interfejsu do precyzyjnego typu danych, na którym działa. Korzyści płynące z typów ogólnych to zwiększone wykorzystanie kodu i bezpieczeństwo typów.[[4]](#_Bibliografia)
* Delegaty i zdarzenia – odpowiednik wskaźników w C++, delegaty przechowują referencję do metody lub grupy metod i nie posiadają implementacji.
* Bogata biblioteka klas BCL (*Base Class Library*), pozwalająca na rozwój aplikacji konsolowych, okienkowych (WinForms, WPF), bazodanowych (ADO.NET), sieciowych (System.Net) oraz internetowych (ASP.NET).

Aplikacja jest tworzona w oparciu o platformę .NET. Po języku C#, platforma .NET jest najważniejszą częścią pracy. .NET to platforma programistyczna stworzona przez firmę Microsoft, obejmująca środowisko uruchomieniowe i wiele różnych bibliotek umożliwiających tworzenie aplikacji.

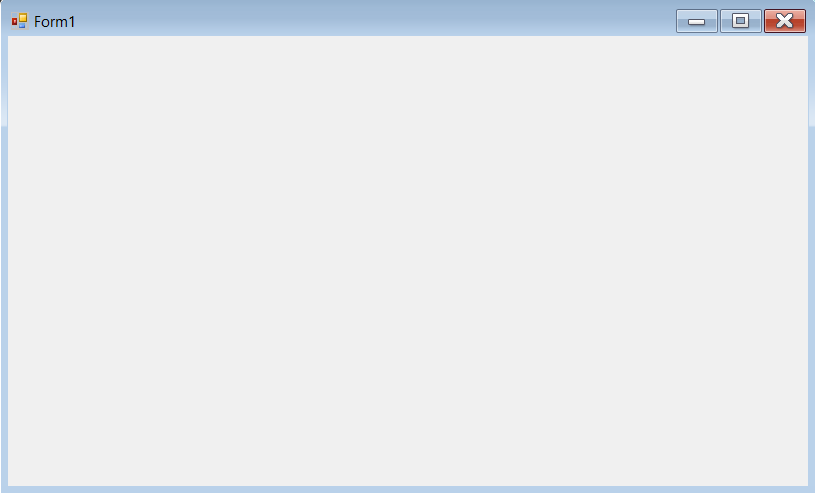


Rys. 2. Komponenty oraz rozwój platformy .NET.

Żródło: Wikipedia[[5]](#_Bibliografia).

Najnowsza wersja .NET Framework to 4.8 (aplikacja używa wersji 4.7.2), a pierwsza wersja została wydana w 2002. Od początku istnienia platforma .NET została rozbudowana o wiele komponentów (Rys. 2). Główne komponenty platformy .NET używane w projekcie to: WinForms, LINQ, i Entity Framework.

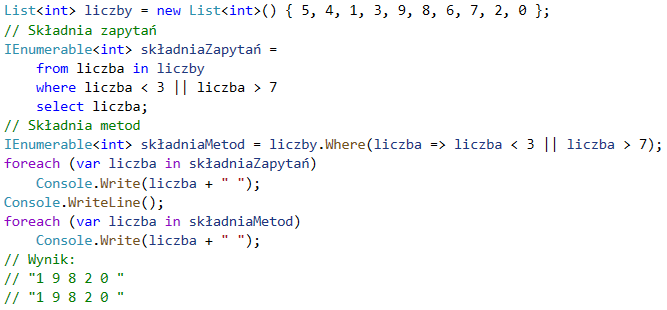
WinForms to graficzny interfejs użytkownika zawarty w .NET Framework, który umożliwia natywny dostęp do elementów interfejsu graficznego systemu Microsoft Windows. Podstawowy wygląd aplikacji z użyciem tej aplikacji jest przedstawiony poniżej (Rys. 2).



Rys. 3. Podstawowy wygląd aplikacji stworzonej z wykorzystaniem biblioteki WinForms.

Źródło: Opracowanie własne

Kolejnym ważnym komponentem platformy .NET użytym w projekcie jest technologia LINQ (ang. *Language INtegrated Query*), która umożliwia zadawanie pytań na obiektach. Dzięki LINQ można łatwo operować na zbiorach danych. Istnieją trzy sposoby na tworzenie zapytań w LINQ: składnia zapytań (ang. *query syntax*), składnia metody (ang. *syntax method*) i połączenie obu składni. W projekcie wykorzystamy LINQ do operacji na zbiorach z bazy danych.



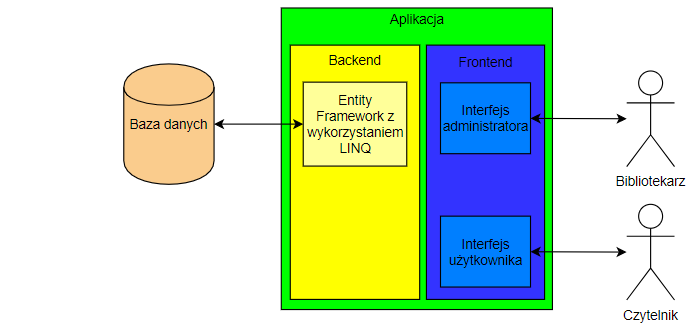
Rys. 4. Przykład wykorzystania LINQ na dwa sposoby: składanie zapytań i składanie metod.

Źródło: Opracowanie własne

W 2008 roku do platformy .NET (do wersji 3.5 SP1) dołączył Entity Framework i był jej integralną częścią aż do 2013 roku, kiedy Entity Framework 6 stał się oddzielną technologią. EF (Entity Framework) jest platformą mapowania obiektowo – relacyjnego (ang. *Object-Relational Mapping – ORM*). Kilka głównych cech EF to:

* Wieloplatformowość – Entity Framework Core może być używany na systemach Windows, Linux i Mac.
* Działania na bazach danych – EF umożliwia korzystanie z LINQ i zapytań SQL do operacji na zbiorach danych w bazach danych.
* Śledzenie zmian – EF śledzi zmiany w instancjach encji, które muszą zostać zgłoszone do bazy danych.
* Zapisywanie zmian – EF wykonuje operacje INSERT, UPDATE, DELETE w zależności od zmiany w bazie danych podczas wywołania metody SaveChanges().
* Migracje – EF oferuje zestaw komend do migracji wywoływanych w CLI (ang. *Command Line Interface*) lub konsoli menadżera pakietów NuGet, aby móc tworzyć i zarządzać schematami baz danych.

## Omówienie struktury aplikacji



Rys. 5. Ogólna struktura aplikacji.

Źródło: Opracowanie własne.

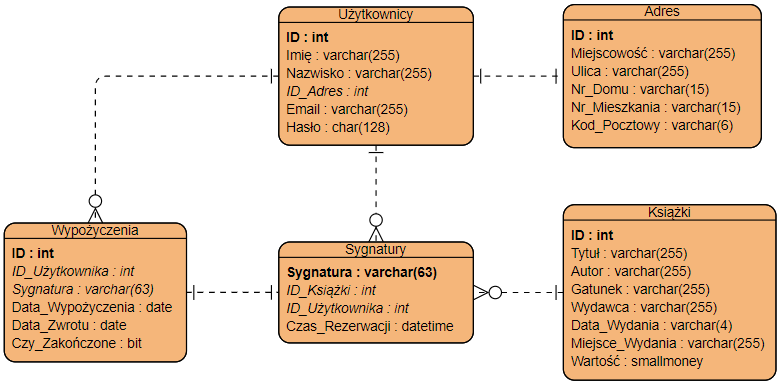
Na stworzenie aplikacji składają się trzy główne zadania. Pierwszym z nich jest stworzenie bazy danych do przechowywania danych z biblioteki. Baza zostanie stworzona za pomocą Entity Framework podejściem „*Code First*”, czyli najpierw stworzymy klasy i relacje między nimi w kodzie projektu, a potem podczas pierwszego uruchomienia aplikacji Entity Framework zainicjuje bazę danych. Entity Framework ułatwia też migracje bazy danych w razie potrzeby wprowadzenia jakichkolwiek zmian w bazie. Wykorzystanie Entity Framework dzięki tym rozwiązaniom pozwala zaoszczędzić czas na tworzeniu danych. Do operacji na zbiorach z bazy danych będzie wykorzystywany LINQ.

Drugim zadaniem jest stworzenie całej logiki aplikacji oraz implementacja wszystkich funkcji pozwalających na korzystanie z aplikacji (patrz. [Cel i ogólny opis projektu](#_Cel_i_ogólny)). To tutaj zostaną zaimplementowane operacje na bazie danych z wykorzystaniem Entity Framework i LINQ.

Trzecim zadaniem jest stworzenie widoku aplikacji zgodnego z systemem Windows 10, który jest podzielony na dwie części: interfejs użytkownika i interfejs administratora. Bibliotekarz ma dostęp do interfejsu administratora, a czytelnik do interfejsu użytkownika.



### Omówienie struktury bazy danych



Rys. 6. Diagram relacji encji (ang. Entities Relationship Diagram).

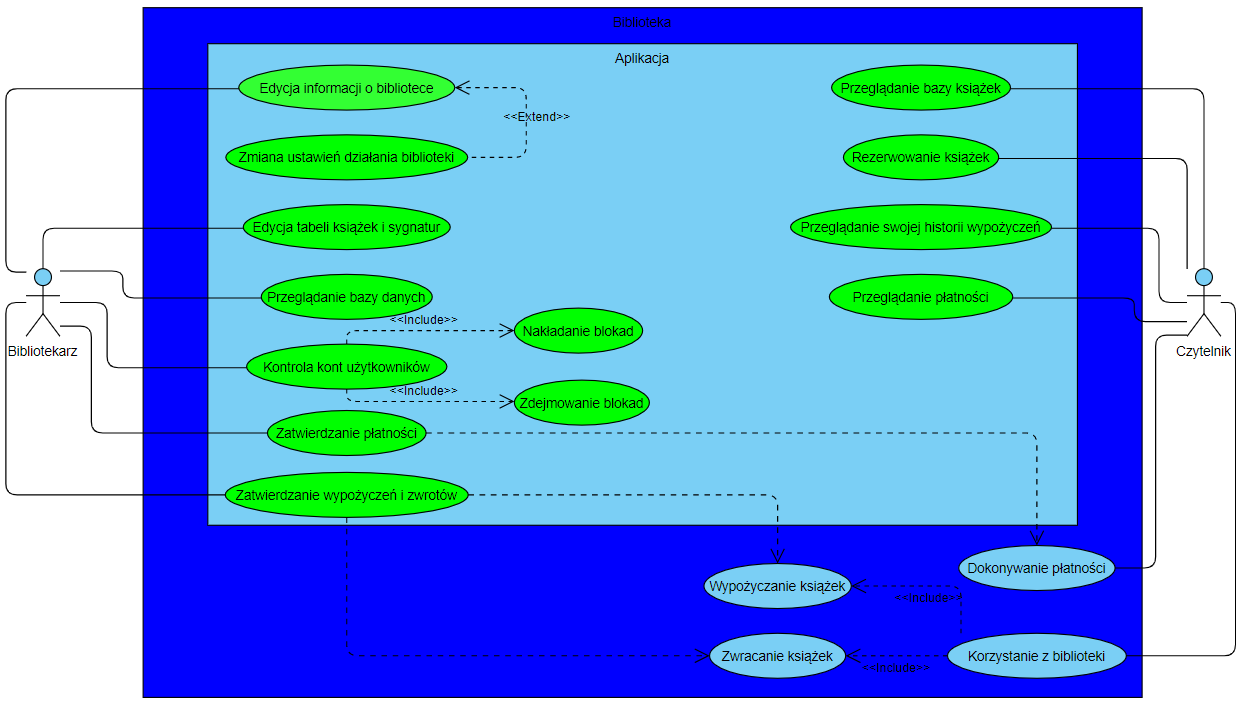
Źródło: Opracowanie własne.

Zanim przejdę do omawiania struktury bazy danych użytej w aplikacji, wyjaśnię kilka pojęć związanych z bazami danych, żeby ułatwić zrozumienie dalszej części tekstu. Tymi pojęciami są:

* Klucz podstawowy (ang. *Primary Key*) – to kolumna (lub zestaw kolumn, wtedy mamy do czynienia z kluczem złożonym) w tabeli, która posiada atrybut klucza podstawowego, a jej wartość jednoznacznie identyfikuje rekord w tabeli. Wartość tej kolumny musi być unikatowa dla każdego rekordu w tabeli.
* Klucz obcy (ang. *Foreign Key*) – to kolumna lub zestaw kolumn w tabeli, która posiada atrybut klucza obcego i oznacza wartość jednoznacznie identyfikującą rekord w innej tabeli będącej w relacji z daną tabelą. Bardzo często jest to klucz podstawowy drugiej z tabel.
* Relacja jeden do jednego (ang. *one to one*) – jest to relacja między tabelami tej samej bazy danych oznaczająca, że jeden rekord z pierwszej tabeli może być powiązany z jednym rekordem z drugiej tabeli.
* Relacja jeden do wielu (ang. *one to many*) – jest to relacja między tabelami tej samej bazy danych oznaczająca, że jeden rekord z pierwszej tabeli może być powiązany z jednym lub wieloma rekordami z drugiej tabeli.
* Relacja wiele do wielu (ang. *many to many*) – jest to relacja między tabelami tej samej bazy danych oznaczająca, że jeden lub wiele rekordów z pierwszej tabeli może być powiązany z jednym lub wieloma rekordami z drugiej tabeli.

Skoro najważniejsze pojęcia zostały wyjaśnione możemy przystąpić do omówienia struktury bazy danych. Baza danych projektu będzie posiadała pięć tabel powiązanych ze sobą relacjami jeden do jednego lub jeden do wielu. Schemat relacji tabel jest przedstawiony na rysunku nr. 4. Tabela Użytkownicy jest w relacji z tabelą Sygnatury i Wypożyczenia jeden do wielu oraz z tabelą Adres jeden do jednego. W tabeli Wypożyczenia kolumna ID to klucz podstawowy (pogrubiona czcionka), a kolumny ID\_Użytkownika i Sygnatura to klucze obce (napisane kursywą). Diagram (Rys. 6) przedstawia też wybór typu danych dla odpowiednich kolumn.

### Omówienie logiki aplikacji



Rys. 7. Diagram przypadków użyć (ang. *Use Case Diagram*)

Źródło: Opracowanie własne.

Aplikacja musi umożliwiać użytkownikom korzystanie z wszystkich wspomnianych w ogólnym opisie projektu funkcji, tak jak przedstawia rysunek nr. 7.

### Omówienie widoku aplikacji

# Bibliografia

1. “Bibliotekarz.NET”, autorski blog Marka Ryńskiego, <https://bibliotekarz.net/> (dostęp 25.11.2020).
2. “*Rozwiązania dla bibliotek - Prolib*”, strona internetowa z ofertami firmy Sygnity Business Solutions, <https://sygnitysbs.pl/oferta/system-dla-bibliotek-prolib> (dostęp 25.11.2020).
3. Klaus Kreft, Angelika Langer. “*After Java and C# - what is next?*”. <https://www.artima.com/weblogs/viewpost.jsp?thread=6543> (dostęp 19.11.2020).
4. “*Typy ogólne w .NET*”, Dokumentacja platformy .NET firmy Microsoft, <https://docs.microsoft.com/pl-pl/dotnet/standard/generics/> (dostęp 19.11.2020).
5. “*.NET Framework component stack*”, Wikipedia, wolna encyclopedia, <https://en.wikipedia.org/wiki/.NET_Framework> (dostęp 19.11.2020).
6. N

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW   
w tym w Archiwum Prac Dyplomowych SGGW

.................................................................

*(czytelny podpis autora pracy)*