



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

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Technologie IoT - Internetu rzeczy

Karol Sawicki
Nr albumu studenta w69845

Aplikacja „Biblioteka” z interfejsem GUI

Prowadzący: mgr inż. Ewa Żesławska

Praca projektowa programowanie obiektowe C#

Rzeszów 2025

Spis treści

Wstęp	4
1 Opis założeń projektu	5
1.1 Cele projektu	5
1.2 Wymagania funkcjonalne i нефункционалне	5
2 Opis struktury projektu	7
2.1 Język, narzędzia i wymagania sprzętowe	7
2.2 Zarządzanie danymi i baza danych	8
2.3 Hierarchia klas i opis funkcjonalności	8
2.4 Opis techniczny struktury aplikacji	10
3 Harmonogram realizacji projektu	11
3.1 Opis harmonogramu realizacji projektu	11
3.2 Problemy i trudności podczas realizacji projektu	12
3.3 Informacje o repozytorium i systemie kontroli wersji	12
4 Prezentacja warstwy użytkowej projektu	13
4.1 Opis warstwy użytkowej	13
4.2 Przegląd interfejsu użytkownika	13
4.3 Podsumowanie	23
5 Podsumowanie	24
5.1 Zrealizowane prace	24
5.2 Planowane prace rozwojowe	24
5.3 Podsumowanie końcowe	25
Bibliografia	26
Spis rysunków	27
Spis tablic	28

Wstęp

Współczesne zarządzanie bibliotekami staje się coraz bardziej złożonym procesem, który wymaga zaawansowanych narzędzi i systemów wspierających codzienną działalność. Biblioteki, zarówno te tradycyjne, jak i nowoczesne, borykają się z problemami związanymi z rejestracją i monitorowaniem wypożyczeń, śledzeniem zwrotów książek, czy zarządzaniem danymi klientów. Problem ten staje się szczególnie widoczny w przypadku większych placówek, gdzie liczba użytkowników oraz zasobów wymaga precyzyjnego planowania i automatyzacji działań.

Projekt „Bibliotekarz” powstał z potrzeby usprawnienia pracy bibliotek poprzez stworzenie systemu wspierającego podstawowe procesy zarządzania. Głównym celem aplikacji jest ułatwienie takich zadań, jak: - dodawanie i edycja danych książek oraz klientów, - rejestrowanie i monitorowanie transakcji wypożyczeń i zwrotów, - przeglądanie danych w czytelnych tabelach oraz - eksportowanie kluczowych informacji do plików Excel, co wspomaga analizę i archiwizację danych.

Aplikacja została stworzona w technologii C#, wykorzystując bibliotekę Windows Forms do interfejsu użytkownika oraz SQL Server jako bazę danych. Dzięki zastosowaniu tych narzędzi system jest intuicyjny w obsłudze i może być łatwo dostosowany do potrzeb użytkowników.

Projekt ten ma na celu nie tylko usprawnienie funkcjonowania bibliotek, ale również przedstawienie pełnego procesu tworzenia aplikacji – od analizy wymagań, przez implementację, aż po testy funkcjonalności. Powstały system jest nie tylko praktycznym rozwiązaniem, ale również może stanowić inspirację dla innych projektów związanych z zarządzaniem zasobami.

Rozdział 1

Opis założeń projektu

1.1 Cele projektu

Zarządzanie biblioteką wymaga sprawnego nadzoru nad zasobami książkowymi, rejestrowania wypożyczeń, śledzenia zwrotów oraz zarządzania danymi użytkowników. Tradycyjne metody prowadzenia dokumentacji często są czasochłonne i podatne na błędy, co prowadzi do problemów organizacyjnych. Celem projektu „Bibliotekarz” jest stworzenie nowoczesnej aplikacji wspierającej zarządzanie biblioteką, która będzie prostym, intuicyjnym i efektywnym narzędziem umożliwiającym pełną kontrolę nad tymi procesami.

Projekt rozwiązuje problem chaotycznego zarządzania, ogranicza ryzyko błędów w dokumentacji oraz usprawnia obsługę klientów biblioteki. Istotność tego problemu wynika z konieczności zapewnienia sprawnego funkcjonowania bibliotek, co jest kluczowe w procesie promowania edukacji i czytelnictwa.

System „Bibliotekarz” zakłada:

- Automatyzację procesu wypożyczania i zwrotu książek.
- Zwiększenie efektywności zarządzania danymi książek oraz użytkowników.
- Możliwość generowania raportów dotyczących transakcji w formacie Excel.
- Czytelny interfejs użytkownika, który minimalizuje czas potrzebny na naukę obsługi.

Realizacja projektu przebiegała według następujących etapów:

1. Analiza wymagań użytkowników biblioteki oraz problemów w tradycyjnym systemie zarządzania.
2. Implementacja kluczowych funkcji systemu, takich jak dodawanie książek, rejestracja klientów, obsługa transakcji wypożyczeń i zwrotów.
3. Stworzenie mechanizmu eksportu danych do formatu Excel w celu łatwiejszej analizy i raportowania.
4. Testowanie systemu w celu wykrycia i eliminacji potencjalnych błędów.

Rezultatem prac jest aplikacja desktopowa, która usprawnia codzienne działania bibliotekarzy i umożliwia szybki dostęp do kluczowych informacji.

1.2 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne

- Możliwość dodawania, edycji i usuwania danych książek oraz klientów.

- Obsługa transakcji wypożyczania i zwrotu książek.
- Rejestrowanie szczegółowych informacji o transakcjach, takich jak daty wypożyczeń i zwrotów.
- Weryfikacja dostępności książek w czasie rzeczywistym.
- Eksport danych o książkach, klientach i transakcjach do plików Excel.
- Wyszukiwanie klientów na podstawie numeru PESEL oraz wyświetlanie ich wypożyczeń.

Wymagania niefunkcjonalne

- Przyjazny i intuicyjny interfejs użytkownika, który wymaga minimalnego szkolenia.
- Wysoka wydajność aplikacji, nawet przy dużej liczbie książek i użytkowników.
- Zgodność z systemem operacyjnym Windows.
- Bezpieczeństwo danych poprzez kontrolę dostępu i walidację danych wejściowych.
- Utrzymywanie integralności bazy danych oraz możliwość odzyskiwania danych w razie awarii.

Rozwinięcie wymagań niefunkcjonalnych

- Aplikacja powinna reagować w ciągu 2 sekund na każdą akcję użytkownika, np. wyszukiwanie książek.
- System powinien być zdolny do obsługi minimum 1000 aktywnych użytkowników.
- Eksport danych do Excela powinien być możliwy w czasie krótszym niż 5 sekund.
- Wszystkie ekrany aplikacji powinny mieć responsywny układ dostosowany do różnych rozdzielczości ekranów.
- System powinien posiadać kopie zapasowe danych, tworzone automatycznie co 24 godziny.

Podsumowując, aplikacja „Bibliotekarz” jest kompleksowym rozwiązaniem, które zaspokaja potrzeby bibliotek, zwiększając efektywność pracy oraz redukując czasochłonność operacji administracyjnych.

Rozdział 2

Opis struktury projektu

Projekt „Bibliotekarz” został zrealizowany jako aplikacja desktopowa stworzona w języku C# przy użyciu platformy .NET Framework. Struktura projektu została zaprojektowana w sposób modułowy, co pozwala na łatwą rozbudowę i utrzymanie systemu. W niniejszym rozdziale omówiono techniczne aspekty projektu, jego architekturę, narzędzia użyte podczas implementacji oraz szczegóły dotyczące zarządzania danymi i bazy danych.

2.1 Język, narzędzia i wymagania sprzętowe

Wykorzystany język programowania

Projekt został napisany w języku C#, który jest wysokopoziomowym językiem programowania zapewniającym wsparcie dla programowania obiektowego. Dzięki swojej wydajności i bogatemu ekosystemowi bibliotek, C# jest doskonałym wyborem do tworzenia aplikacji desktopowych.

Narzędzia użyte w projekcie

- **Visual Studio 2022** – główne środowisko programistyczne (IDE) użyte do tworzenia i debugowania aplikacji.
- **SQL Server Management Studio (SSMS)** – narzędzie do zarządzania bazą danych.
- **EPPlus** – biblioteka .NET umożliwiająca generowanie i modyfikację plików Excel w celu eksportu danych.
- **GitHub** – system kontroli wersji użyty do zarządzania kodem i współpracy zespołowej.

Minimalne wymagania sprzętowe

- Procesor: minimum 1 GHz.
- Pamięć RAM: minimum 2 GB.
- Dysk twardy: co najmniej 100 MB wolnego miejsca.
- System operacyjny: Windows 7 lub nowszy.
- Zainstalowane środowisko .NET Framework w wersji 4.7.2 lub wyższej.
- SQL Server 2019 lub kompatybilna wersja bazy danych SQL.

2.2 Zarządzanie danymi i baza danych

Dane w systemie „Bibliotekarz” są zarządzane przy użyciu relacyjnej bazy danych SQL Server. Struktura bazy danych została zaprojektowana tak, aby przechowywać kluczowe informacje związane z funkcjonowaniem biblioteki, takie jak:

- **Tabela Books** – zawiera informacje o książkach, w tym identyfikator, tytuł, autora, datę wydania, liczbę egzemplarzy oraz liczbę dostępnych książek.
- **Tabela Clients** – przechowuje dane o czytelnikach, takie jak imię, nazwisko, numer PESEL, numer telefonu i adres zamieszkania.
- **Tabela Transactions** – rejestruje szczegóły wypożyczeń i zwrotów książek, takie jak identyfikator transakcji, identyfikator klienta, identyfikator książki, datę wypożyczenia i datę zwrotu.

Dane w systemie są przetwarzane za pomocą poleceń SQL, takich jak SELECT, INSERT, UPDATE i DELETE, które są realizowane w warstwie logicznej aplikacji.

2.3 Hierarchia klas i opis funkcjonalności

Projekt został podzielony na klasy, które odzwierciedlają główne elementy systemu. Poniżej przedstawiono najważniejsze klasy wraz z ich opisem:

Klasa **LibraryManager**

Jest to klasa statyczna odpowiedzialna za zarządzanie logiką biznesową aplikacji oraz komunikację z bazą danych.

- **Metoda AddClient (Client client)** – Dodaje nowego klienta do bazy danych.
- **Metoda GetBooks ()** – Zwraca listę wszystkich książek dostępnych w bibliotece.
- **Metoda AddTransaction (int clientId, long bookId, DateTime loanDate)** – Rejestruje nową transakcję wypożyczenia książki.
- **Metoda ReturnBook (int transactionId, DateTime returnDate)** – Aktualizuje transakcję, rejestrując datę zwrotu książki.
- **Metoda ExportTableToExcel (string tableName, string filePath)** – Eksportuje listę książek do pliku Excel.

Klasa **Book**

Reprezentuje książkę w systemie. Zawiera właściwości:

- **BookID** – unikalny identyfikator książki.
- **BookName** – tytuł książki.
- **Author** – autor książki.
- **ReleaseDate** – data wydania książki.
- **NumberOfBooks** – całkowita liczba egzemplarzy książki.
- **AvailableBooks** – liczba dostępnych egzemplarzy książki.

Klasa **Client**

Reprezentuje klienta biblioteki. Zawiera właściwości:

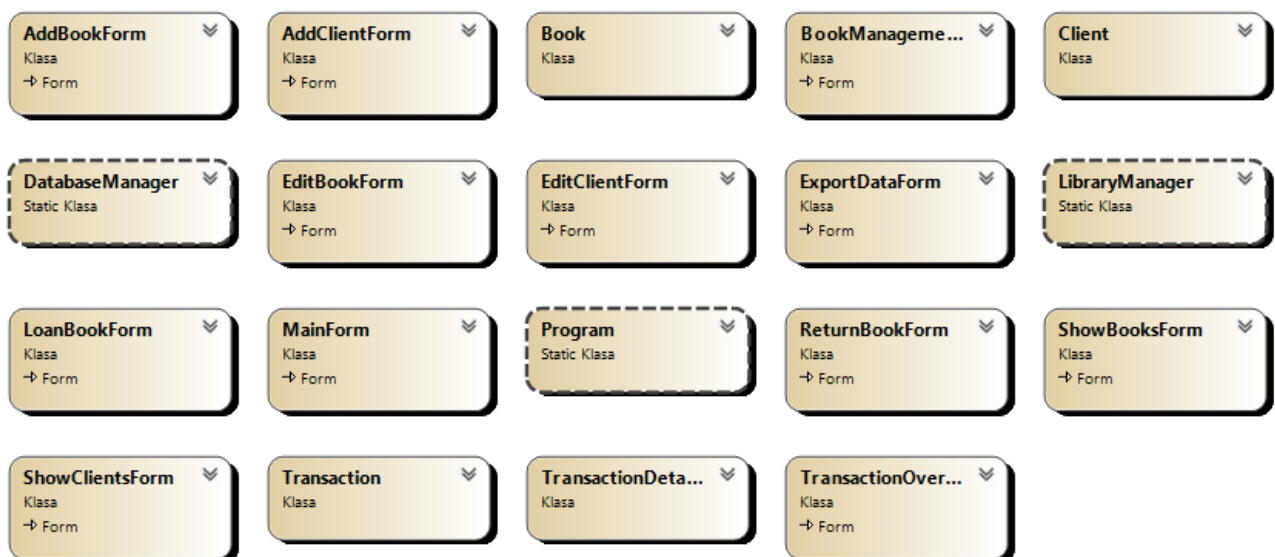
- `ClientID` – unikalny identyfikator klienta.
- `Name` – imię klienta.
- `Surname` – nazwisko klienta.
- `PESEL` – numer PESEL klienta.
- `TelephoneNumber` – numer telefonu klienta.
- `HomeAddress` – adres zamieszkania klienta.

Klasa **Transaction**

Reprezentuje transakcję wypożyczenia lub zwrotu książki. Zawiera właściwości:

- `TransactionID` – unikalny identyfikator transakcji.
- `ClientID` – identyfikator klienta.
- `BookID` – identyfikator książki.
- `LoanDate` – data wypożyczenia książki.
- `ReturnDate` – data zwrotu książki (opcjonalna).

Diagram klas:



Rysunek 2.1: Diagram klas wygenerowane w visual studio”.

2.4 Opis techniczny struktury aplikacji

Struktura aplikacji została podzielona na trzy główne warstwy:

- **Warstwa prezentacji (UI):** Odpowiada za interakcję z użytkownikiem i zawiera formularze, takie jak `MainForm`, `LoanBookForm` i `ReturnBookForm`.
- **Warstwa logiki biznesowej:** Klasa `LibraryManager` obsługuje procesy biznesowe aplikacji.
- **Warstwa dostępu do danych:** Obsługuje komunikację z bazą danych za pomocą kwerend SQL i zapewnia trwałość danych.

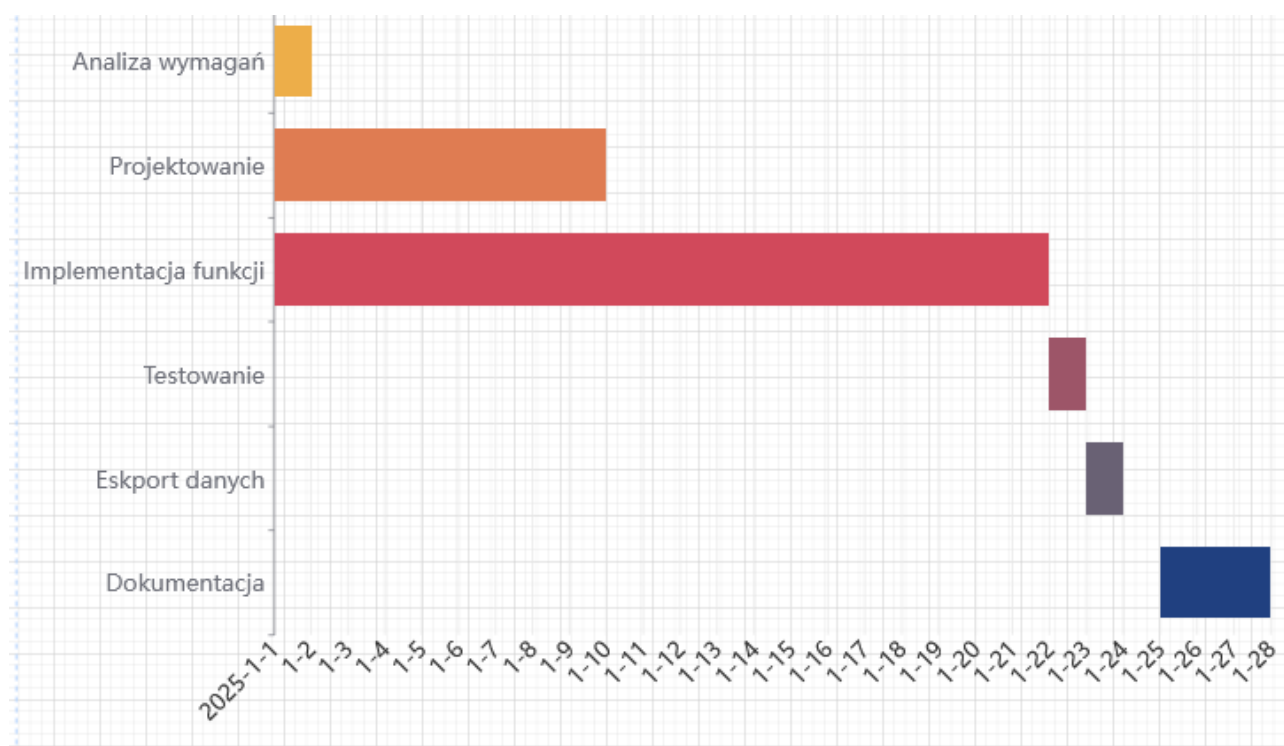
Aplikacja została zaprojektowana w sposób pozwalający na łatwe rozbudowywanie o nowe funkcjonalności, co czyni ją elastycznym narzędziem do zarządzania biblioteką.

Rozdział 3

Harmonogram realizacji projektu

3.1 Opis harmonogramu realizacji projektu

Realizacja projektu „Bibliotekarz” została zaplanowana i wykonana zgodnie z harmonogramem opartym na diagramie Gantta. Diagram ten przedstawia podział projektu na poszczególne etapy, takie jak analiza wymagań, projektowanie, implementacja oraz testowanie. Każdy etap został dodatkowo podzielony na mniejsze zadania, co pozwoliło na lepsze monitorowanie postępów prac.



Rysunek 3.1: Diagram Gantta przedstawiający harmonogram realizacji projektu „Bibliotekarz”.

Na podstawie diagramu Gantta można wyróżnić następujące etapy realizacji projektu:

- **Analiza wymagań:** Przeprowadzono wstępną analizę potrzeb użytkowników i zaprojektowano założenia funkcjonalne oraz нефункционалне systemu.
- **Projektowanie:** Stworzono strukturę projektu, zaprojektowano bazę danych oraz interfejs użytkownika.
- **Implementacja:** Zrealizowano główne funkcjonalności systemu, takie jak zarządzanie książkami, klientami oraz transakcjami.

- **Testowanie:** Przeprowadzono testy funkcjonalne i wydajnościowe aplikacji oraz naprawiono zidentyfikowane błędy.
- **Eksport danych:** Dodano funkcję eksportu danych do formatu Excel.
- **Dokumentacja:** Opracowano dokumentację użytkownika oraz opis techniczny projektu.

3.2 Problemy i trudności podczas realizacji projektu

Podczas realizacji projektu napotkano kilka problemów i trudności, które wymagały rozwiązania:

- **Zarządzanie dostępnością książek:** Problem z synchronizacją liczby dostępnych książek po zwrocie został rozwiązany przez dodanie odpowiedniej logiki w metodzie `ReturnBook()`.
- **Obsługa identycznych imion i nazwisk klientów:** Wprowadzono dodatkowe pole PESEL w interfejsie wypożyczeń, co pozwoliło na jednoznaczną identyfikację klienta.
- **Problemy z eksportem danych:** Konfiguracja biblioteki `EPPlus` wymagała instalacji odpowiednich zależności i aktualizacji wersji środowiska `.NET`.
- **Kompatybilność z bazą danych:** W trakcie testów wystąpiły problemy z typami danych w tabelach SQL, które zostały rozwiązane poprzez migrację do właściwego typu `BIGINT` dla identyfikatorów książek.

3.3 Informacje o repozytorium i systemie kontroli wersji

Kod źródłowy projektu jest dostępny publicznie w repozytorium GitHub pod adresem:

<https://github.com/WildGenie7/BiBliotekarz>

Do zarządzania wersjami projektu wykorzystano system kontroli wersji **Git**. Repozytorium pozwoliło na:

- śledzenie zmian w kodzie i łatwe cofanie się do wcześniejszych wersji w przypadku błędów,
- współpracę nad kodem, która zapewniała płynne wprowadzanie nowych funkcjonalności,
- automatyczną synchronizację zmian między lokalnymi kopiami a wersją w chmurze.

Każda zmiana w projekcie była dokumentowana w postaci opisowych komunikatów, co pozwoliło na zachowanie czytelności historii zmian. Repozytorium będzie dostępne przez minimum rok od zakończenia projektu, co umożliwi łatwy dostęp do kodu w celach edukacyjnych lub ewentualnych poprawek.

Rozdział 4

Prezentacja warstwy użytkowej projektu

4.1 Opis warstwy użytkowej

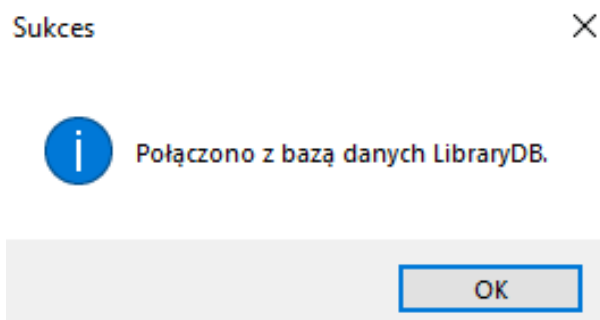
Warstwa użytkowa projektu została zaprojektowana w sposób intuicyjny i ergonomiczny, aby umożliwić użytkownikowi wygodne korzystanie z aplikacji. Interfejs jest podzielony na główne moduły, dostępne za pomocą przycisków w oknie głównym aplikacji. Każda funkcjonalność jest wyraźnie oznaczona i zorganizowana tak, aby minimalizować liczbę kliknięć potrzebnych do wykonania danej operacji.

Wszystkie formularze zostały wyposażone w podstawowe elementy interakcji, takie jak listy rozwijane, pola tekstowe czy przyciski. Aplikacja prowadzi użytkownika krok po kroku przez procesy takie jak wypożyczenie książki, jej zwrot czy zarządzanie bazą klientów.

4.2 Przegląd interfejsu użytkownika

Okno połączenia z bazą danych

Po uruchomieniu programu, wyskoczy komunikat czy udało się połączyć z bazą danych, jest to niezbędne do prawidłowego działania aplikacji.

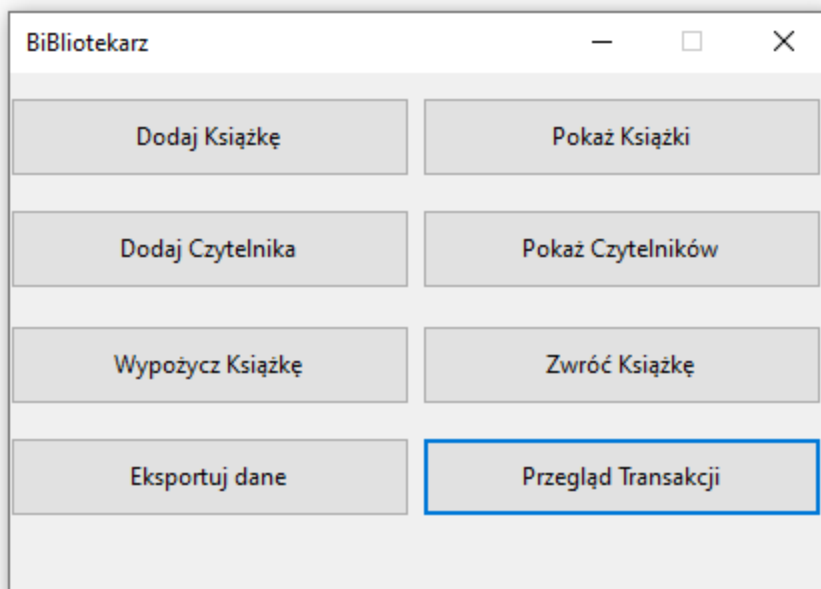


Rysunek 4.1: Komunikat połączenia z bazą danych.

Główne okno aplikacji

Główne okno aplikacji pozwala na dostęp do wszystkich kluczowych funkcji. Widoczne opcje:

- Dodaj Książkę – formularz dodawania nowej książki.
- Pokaż Książki – pozwala na edycję oraz przegląd listy książek.
- Dodaj Czytelnika - formularz dodawania nowego czytelnika.
- Pokaż Czytelników - pozwala na edycję oraz przegląd listy czytelników.
- Wypożycz książkę – formularz wypożyczenia.
- Zwróć Książkę - formularz do obsługi zwrotów.
- Eksportuj dane – umożliwia wyeksportowanie danych książek, klientów i transakcji do pliku Excel.
- Przegląd transakcji – wyświetla listę aktywnych i zakończonych transakcji.



Rysunek 4.2: Widok głównego okna aplikacji.

Formularz wypożyczenia książki

Formularz umożliwia wybór klienta oraz książki, którą chce wypożyczyć. Wybór klienta ułatwiono poprzez automatyczne wyświetlenie numeru PESEL, co pozwala uniknąć pomyłek w przypadku użytkowników o takich samych imionach i nazwiskach.

Rysunek przedstawiający formularz wypożyczenia książki:

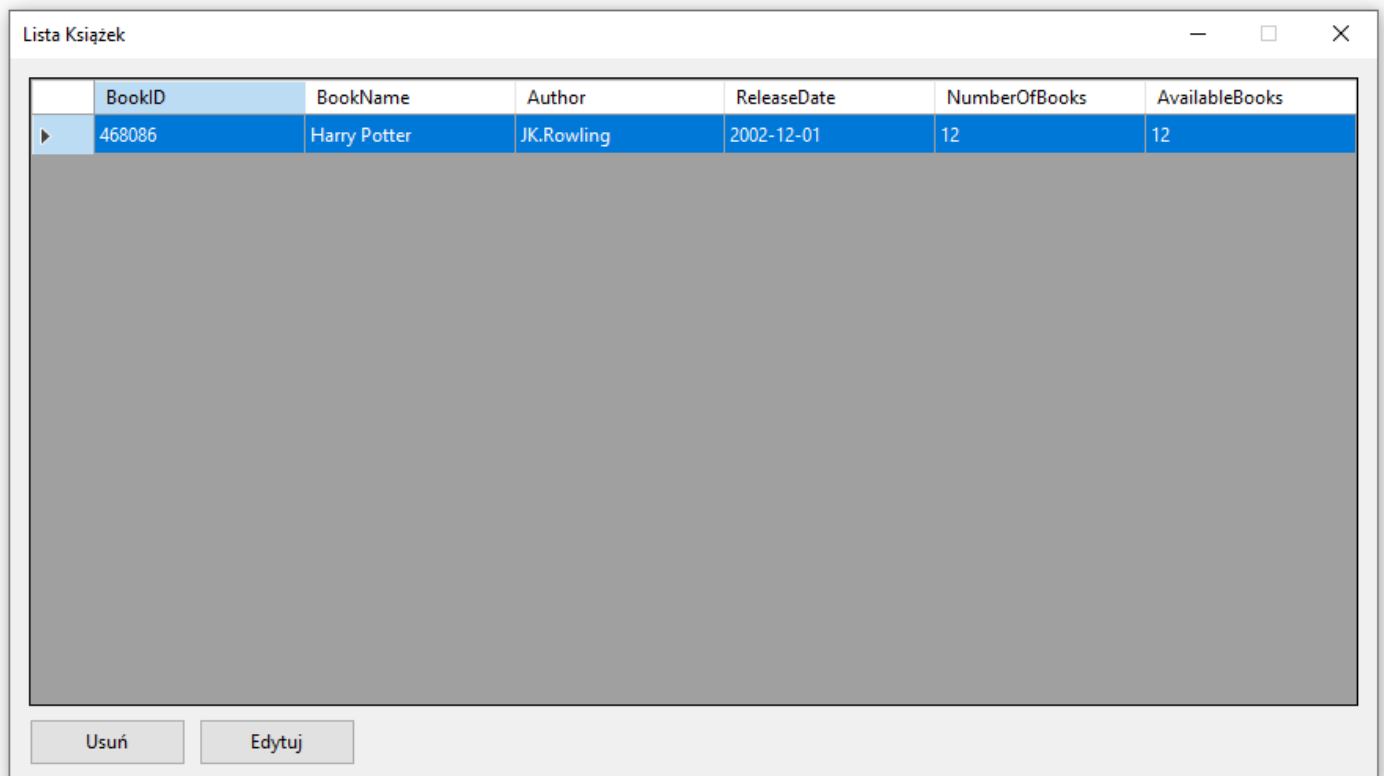
The image shows a graphical user interface window titled "Dodaj Książkę". It contains the following elements:

- Input field for "Nazwa Książki" (Book Name).
- Input field for "Autor" (Author).
- Input field for "Data Wydania" (Publication Date) with a "YYYY-MM-DD" format label to its right.
- Input field for "Liczba Egzemplarzy" (Number of Copies).
- Input field for "ID Książki" (Book ID).
- A "Zapisz" (Save) button at the bottom.

Rysunek 4.3: Widok formularza wypożyczenia książki.

Formularz pokazania książki

Formularz pozwala na pokazanie listy książek z bazy danych, można także je usunąć lub edytować i wszystkie zmiany zostają zapisane w bazie danych. W przypadku braku książek jest wyświetlany odpowiedni komunikat.



The screenshot shows a web application window titled "Lista Książek". Inside the window, there is a table with the following columns: BookID, BookName, Author, ReleaseDate, NumberOfBooks, and AvailableBooks. The first row of data shows a book with ID 468086, titled "Harry Potter" by "JK.Rowling", released on "2002-12-01", with 12 books in total and 12 available. Below the table, there are two buttons: "Usuń" (Delete) and "Edytuj" (Edit).

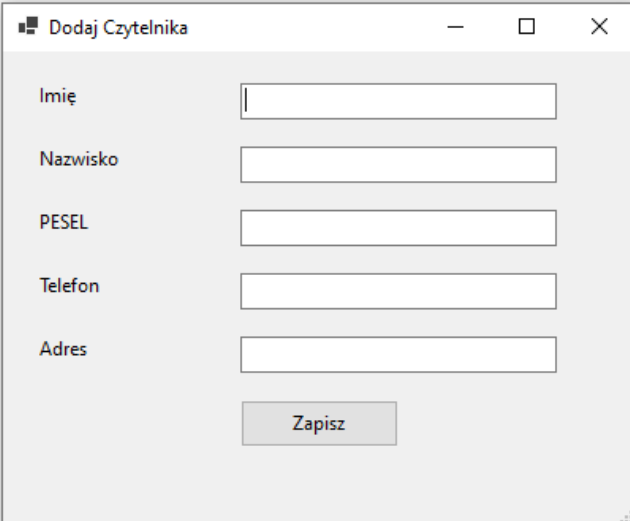
	BookID	BookName	Author	ReleaseDate	NumberOfBooks	AvailableBooks
▶	468086	Harry Potter	JK.Rowling	2002-12-01	12	12

Usuń Edytuj

Rysunek 4.4: Widok formularza pokazania książek.

Formularz dodania czytelnika

Formularz ten pozwala na dodanie nowego czytelnika do bazy danych, posiada odpowiednie walidacje, jak np. 9 cyfr numeru pesel.

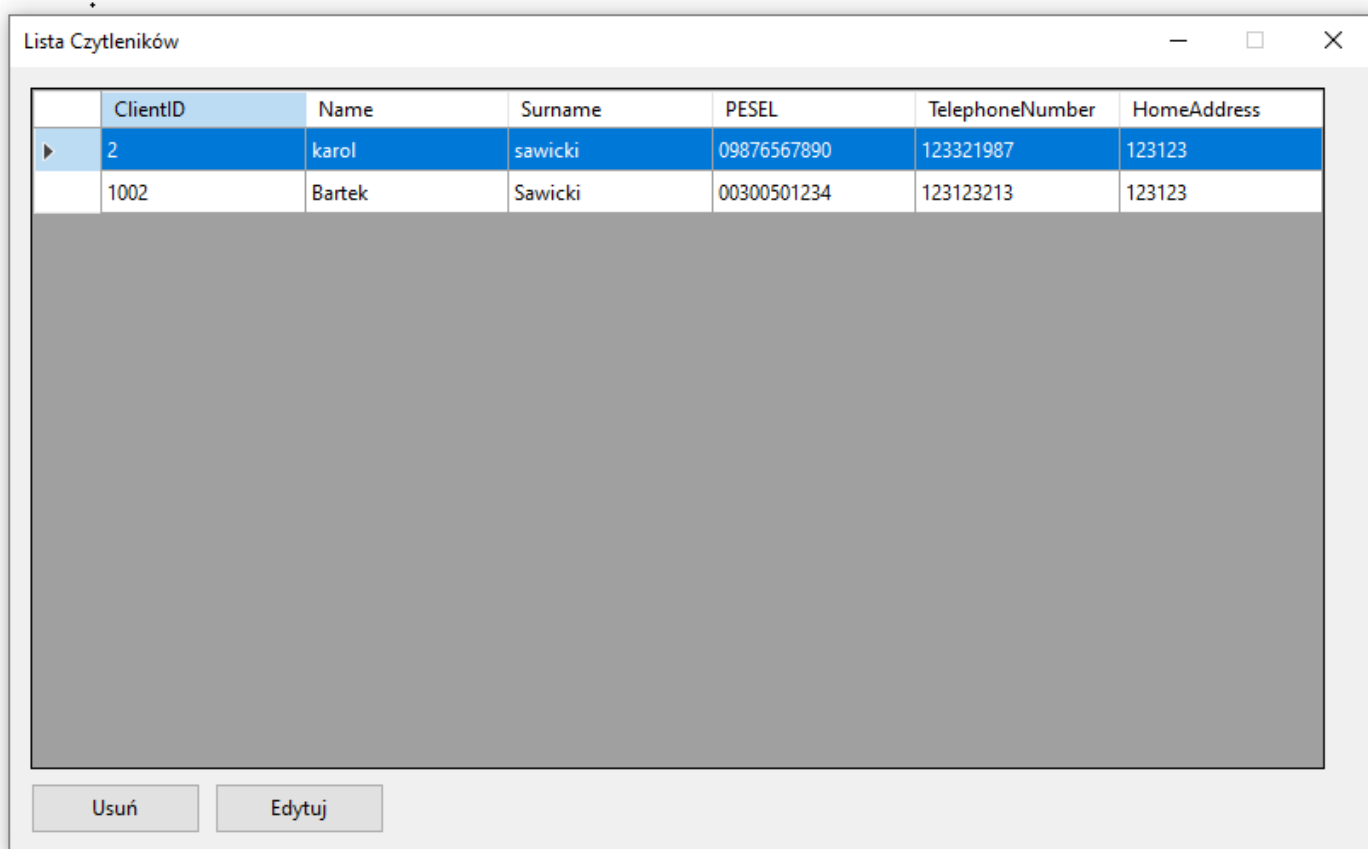


The image shows a Windows-style application window titled "Dodaj Czytelnika". The window has a standard title bar with a maximize button (square icon) and a close button (X icon). The main area of the window is light gray and contains five text input fields arranged vertically. Each field is preceded by a label: "Imię", "Nazwisko", "PESEL", "Telefon", and "Adres". Below these fields is a single button labeled "Zapisz". The form is designed for adding a new reader to a database, with specific validation rules mentioned in the text, such as a 9-digit PESEL number.

Rysunek 4.5: Widok formularza dodania czytelnika.

Formularz pokazania czytelników

Formularz pozwala na pokazanie listy użytkowników z bazy danych, można także je usunąć lub edytować i wszystkie zmiany zostają zapisane w bazie danych. W przypadku braku użytkowników jest wyświetlany odpowiedni komunikat.



	ClientID	Name	Surname	PESEL	TelephoneNumber	HomeAddress
▶	2	karol	sawicki	09876567890	123321987	123123
	1002	Bartek	Sawicki	00300501234	123123213	123123

Usuń Edytuj

Rysunek 4.6: Widok formularza pokazania czytelników.

Formularz wypożyczenia książki

Formularz pozwala na wybranie czytelnika, oraz nazwy książki do wypożyczenia. Od razu pokazuje się pesel do weryfikacji (2 osoby mogą nazywać się tak samo, ale pesel jest unikalny. Po kliknięciu wypożycz jest rejestrowana transakcja w bazie danych z datą dzisiejszą. Jest także odejmowana książka z dostępnych w bazie.

Rysunek 4.7: Widok formularza wypożyczenia książki.

Formularz zwrotu książki

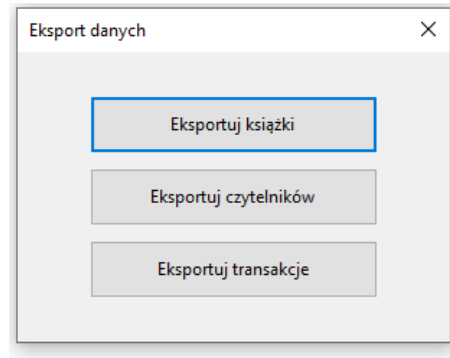
Formularz pozwala na wyszukanie czytelnika po numerze pesel, widać tam wypożyczone przez danego czytelnika książki. Kiedy zostały wypożyczone, do kiedy mają być zwrócone (jest 30dni), jest także pokazana naliczona kara oraz numer transakcji z bazy danych.

	TransactionID	BookTitle	LoanDate	DueDate	Fine
▶	6007	Harry Potter	24.01.2025	23.02.2025	0

Rysunek 4.8: Widok formularza zwrotu książek.

Formularz eksportu danych

Formularz pozwala na eksport danych z bazy danych SQL do formatu CSV, do np. dalszej analizy.

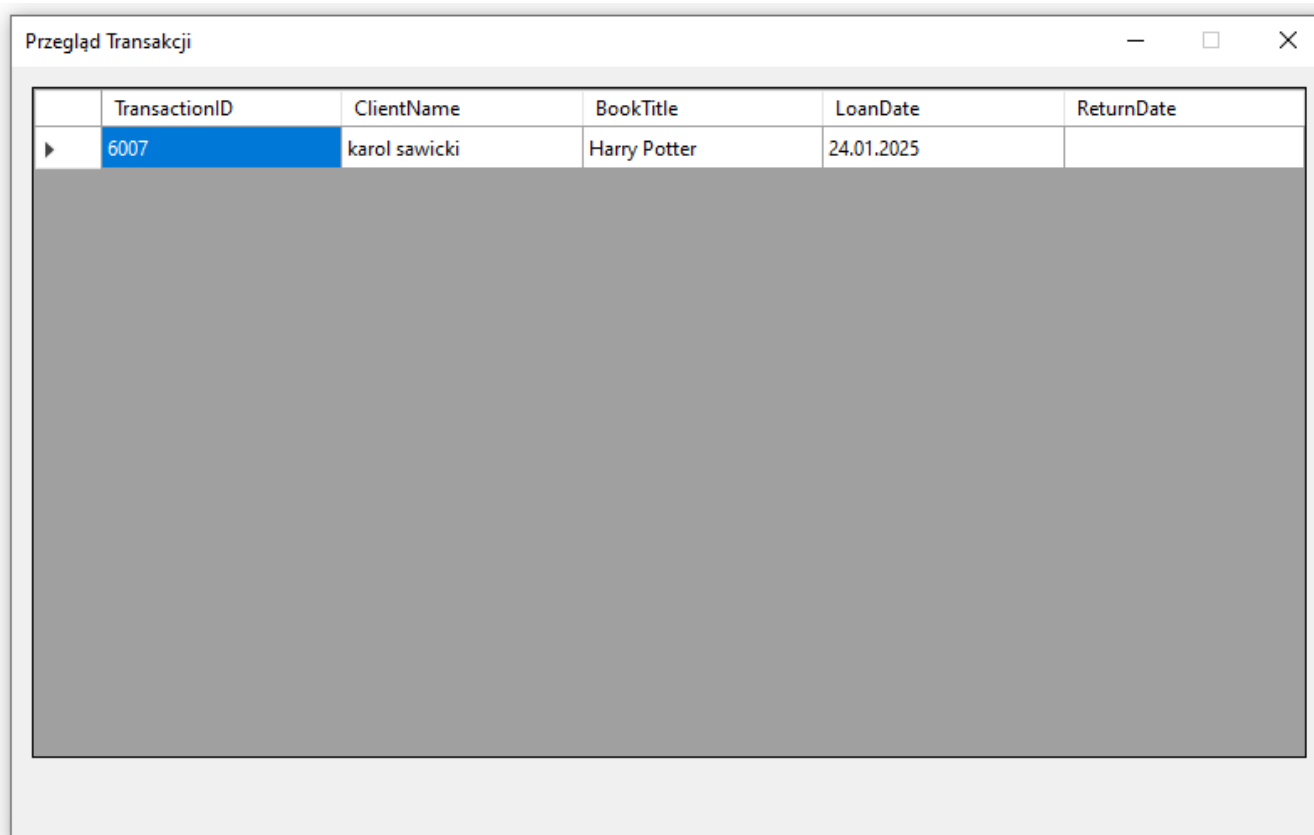


Rysunek 4.9: Widok formularza eksportu danych.

Przegląd transakcji

Formularz umożliwia przegląd wszystkich transakcji zapisanych w systemie. Informacje takie jak nazwa książki, klient, data wypożyczenia i zwrotu, a także ewentualne kary za zwłokę, są czytelnie wyświetlane w tabeli.

Rysunek przedstawiający przegląd transakcji:



The screenshot shows a window titled "Przegląd Transakcji" with a table containing transaction records. The table has five columns: TransactionID, ClientName, BookTitle, LoanDate, and ReturnDate. The first row is highlighted in blue, showing TransactionID 6007, ClientName karol sawicki, BookTitle Harry Potter, and LoanDate 24.01.2025. The ReturnDate column is empty. Below the table is a large gray rectangular area.

	TransactionID	ClientName	BookTitle	LoanDate	ReturnDate
▶	6007	karol sawicki	Harry Potter	24.01.2025	

Rysunek 4.10: Widok przeglądu transakcji.

4.3 Podsumowanie

Aplikacja „Bibliotekarz” została zaprojektowana w sposób maksymalnie przyjazny użytkownikowi, zapewniając przejrzysty i intuicyjny interfejs. Dzięki odpowiedniemu rozmieszczeniu funkcji oraz mechanizmom takim jak walidacja danych czy wyszukiwanie po numerze PESEL, aplikacja minimalizuje możliwość popełnienia błędów przez użytkownika.

Rozdział 5

Podsumowanie

5.1 Zrealizowane prace

Projekt „Bibliotekarz” został pomyślnie ukończony i zrealizowano wszystkie założone cele. W ramach prac:

- Zaprojektowano i zaimplementowano aplikację do zarządzania biblioteką, umożliwiającą wykonywanie takich operacji jak dodawanie, edycja i usuwanie książek oraz klientów.
- Wdrożono funkcjonalności związane z wypożyczaniem i zwracaniem książek, z uwzględnieniem walidacji danych i automatycznego obliczania kar za nieterminowy zwrot.
- Zaimplementowano moduł przeglądu transakcji, pozwalający na pełną kontrolę nad historią wypożyczeń w bibliotece.
- Dodano funkcjonalność eksportu danych (książek, klientów oraz transakcji) do plików Excel, co ułatwia dalsze przetwarzanie informacji poza aplikacją.
- Stworzono czytelny i intuicyjny interfejs użytkownika, który umożliwia łatwe poruszanie się po systemie nawet osobom z niewielkim doświadczeniem technicznym.

5.2 Planowane prace rozwojowe

W przyszłości planowane są następujące rozszerzenia i udoskonalenia aplikacji:

- Wdrożenie systemu wielodostępowego z uwierzytelnianiem użytkowników, co pozwoli na przypisanie różnych poziomów dostępu (np. administrator, bibliotekarz, czytelnik).
- Dodanie mechanizmu importu danych z plików Excel, umożliwiającego szybkie załadowanie dużych baz książek i klientów.
- Rozbudowa interfejsu użytkownika o dodatkowe funkcje analityczne, takie jak raporty dotyczące najczęściej wypożyczanych książek czy zestawienia aktywności użytkowników.
- Zaimplementowanie powiadomień e-mailowych, które informowałyby użytkowników o zbliżających się terminach zwrotu książek lub zaległych płatnościach.
- Przeniesienie aplikacji na platformę internetową, co pozwoli na jej używanie w chmurze, z możliwością dostępu z dowolnego urządzenia z dostępem do Internetu.

5.3 Podsumowanie końcowe

Projekt „Bibliotekarz” jest funkcjonalnym i gotowym do użycia systemem, który odpowiada na potrzeby małych bibliotek. Aplikacja spełnia swoje główne zadanie, jakim jest uproszczenie zarządzania wypożyczaniem książek oraz ich zwrotem, a także udostępnianie kompleksowych danych o klientach i transakcjach.

Wdrożenie systemu w praktyce może przyczynić się do poprawy organizacji pracy w bibliotekach oraz zwiększenia komfortu zarówno pracowników, jak i użytkowników. Dzięki solidnym podstawom technologicznym oraz możliwości dalszego rozwoju, projekt stanowi obiecującą bazę dla kolejnych ulepszeń i adaptacji do nowych potrzeb.

Bibliografia

Spis rysunków

2.1	Diagram klas wygenerowane w visual studio".	9
3.1	Diagram Gantta przedstawiający harmonogram realizacji projektu „Bibliotekarz”.	11
4.1	Komunikat połączenia z bazą danych.	13
4.2	Widok głównego okna aplikacji.	14
4.3	Widok formularza wypożyczenia książki.	15
4.4	Widok formularza pokazania książek.	16
4.5	Widok formularza dodania czytelnika.	17
4.6	Widok formularza pokazania czytelników.	18
4.7	Widok formularza wypożyczenia książki.	19
4.8	Widok formularza zwrotu książek.	20
4.9	Widok formularza eksportu danych.	21
4.10	Widok przeglądu transakcji.	22

Spis tabel