

Programowanie komponentowe

**Projekt – sprawozdanie**

***BIBLIOTEKA***

Dominik Cegiełka 224478

Kamil Zarych 224546

Informatyka, sem. IV, studia I st. niestacjonarne

2019/ 2020

## Spis treści

1. Wstęp .....	3
2. Wymagania i konfiguracja aplikacji .....	3
3. Najważniejsze klasy warstwy danych .....	4
4. Najważniejsze klasy warstwy logiki.....	5
5. Najważniejsze klasy warstwy interfejsu.....	6
6. Zrzuty ekranu najistotniejszych widoków działającej aplikacji .....	6
6.1 Główne okno programu .....	6
6.2 Zakładka „Czytelnicy” – zarządzanie czytelnikami .....	9
6.3 Zakładka „Wypożyczenia” – zarządzanie wypożyczeniami .....	10
6.4 Zakładka „Zwroty” – zarządzanie zwrotami książek i przypomnieniami .....	11
6.5 Okienko „O programie” z opisem klawiszy funkcyjnych .....	12
6.6 Dodatkowe funkcjonalności .....	12
7. Postać bazy danych .....	13
8. Podsumowanie .....	14

## 1. Wstęp

Celem zadania było napisanie aplikacji w Javie z graficznym i terminalowym interfejsem użytkownika. Funkcjonalność aplikacji umożliwia obsługę systemu biblioteki i została zaprojektowana w modelu trójwarstwowym MVC (Model-View-Controller). Dane przechowywane są w bazie danych oraz mogą być zapisane w formacie XML.

Poszczególne funkcjonalności aplikacji:

- Dodawanie, edycja oraz usuwanie książek,
- Dodawanie, edycja oraz usuwanie czytelników,
- Dodawanie oraz edytowanie wypożyczeń,
- Ustawianie przypomnień na określony czas przed datą zwrotu książki,
- Graficzny kalendarz zawierający informacje o aktualnych i archiwalnych wypożyczeniach oraz przypomnienia o zwrocie książki,
- Import/eksport danych do/z pliku XML zawierający informacje o całej strukturze biblioteki,
- Filtrowanie książek, czytelników, wypożyczeń,
- Możliwość spersonalizowania wyglądu aplikacji np. logo, motyw (z zapisem ustawień w pliku XML).
- Wyświetlenie okienka dialogowego „O programie”,
- Walidacja poprawności wprowadzonych danych (np. brak tytułu książki, numer ISBN, numer telefonu, adres e-mail),
- Obsługa przy pomocy myszy oraz klawiatury (skrót klawiaturowy typu CTRL+F, tabulator do zmiany aktywnego pola formularza)

## 2. Wymagania i konfiguracja aplikacji

Wymagania programowe:

- System Windows 10
- Java SE 11
- [JavaFX Windows SDK 11.0.2](#)
- Baza danych MySQL

Aby uruchomić program zmienna środowiskowa `%JAVA_HOME%` powinna wskazywać na katalog z wymaganą wersją Javy oraz zmienna `%PATH_TO_FX%` na katalog `lib` z SDK JavaFX. Ewentualnie możliwa jest ręczne ustawienie ścieżek w plikach wsadowym `run.bat` / `run_tui.bat`.

Po pierwszym uruchomieniu aplikacji wygenerowany zostanie plik `app.properties` z domyślnymi ustawieniami programu. Plik konfiguracyjny zawiera następujące atrybuty:

- **applcon** – ścieżka do ikony programu
- **hostname** – adres/host bazy danych

- **password** – hasło do bazy
- **databaseName** – nazwa bazy
- **port** – numer portu
- **darkTheme** – czy aktywny ciemny motyw (*true/false*)
- **serverTimezone** – strefa czasowa
- **appTitle** – wyświetlany tytuł aplikacji
- **username** – nazwa użytkownika bazy

W przypadku braku połączenia z bazą danych wyświetlony zostanie odpowiedni komunikat z błędem. Praca w aplikacji bez połączenia z bazą danych jest możliwa, ale przed zamknięciem aplikacji należy [wyeksportować dane do pliku XML](#) aby nie zostały utracone.

Po poprawnym skonfigurowaniu parametrów do połączenia z bazą danych, podczas uruchomienia programu po połączeniu z serwerem bazodanowym sprawdzane jest istnienie wymaganych tabel w bazie danych. Jeśli tabele nie istnieją lub nie ma wszystkich wymaganych przez program zostają stworzone. Możliwe jest również ręczne stworzenie wymaganych tabel przy użyciu pliku *CREATETABLES.sql*.

### 3. Najważniejsze klasy warstwy danych

Module `org.pk.library`

Package **`org.pk.library.model`**

Pakiet z modelem danych aplikacji.

Class Summary	
Class	Description
<b>Book</b>	Klasa reprezentująca książkę.
<b>Library</b>	Klasa reprezentująca bibliotekę.
<b>LibraryDB</b>	Klasa odpowiedzialna za komunikację z bazą danych.
<b>LibraryXML</b>	Klasa odpowiedzialna za reprezentację danych w formacie XML.
<b>Reader</b>	Klasa reprezentująca czytelnika.
<b>Rent</b>	Klasa reprezentująca wypożyczenie książki.
<b>RentalReminder</b>	Klasa reprezentująca przypomnienie o wypożyczeniu książki.

*Ilustracja 1: Pakiet warstwy danych z dokumentacji JavaDoc.*

Warstwa danych (model) posiada metody pozwalające na tworzenie obiektów typu Książka, Klient, Wypożyczenie oraz Przypomnienie. Klasy te posiadają gettery do pobrania potrzebnych danych, setery do wprowadzenia ewentualnych poprawek oraz metodę `toString()`. Przetwarzaniem danych obiektów zajmuje się warstwa logiki.

## 4. Najważniejsze klasy warstwy logiki

Module org.pk.library  
Package org.pk.library.controller  
**Class Controller**  
  
java.lang.Object  
org.pk.library.controller.Controller

public class Controller  
extends java.lang.Object  
Kontroler

Field Summary

Fields

Modifier and Type	Field	Description
(package private) java.util.regex.Pattern	emailAddressPattern	Wzorzec walidacyjny dla adresu email.
(package private) java.util.regex.Pattern	isbnPattern	Wzorzec walidacyjny ISBN.
(package private) Library	library	Klasa kontenerowa przechowująca strukturę danych biblioteki.
(package private) LibraryDB	libraryDB	Klasa umożliwiająca operacje na bazie danych.
(package private) java.util.regex.Pattern	phoneNumberPattern	Wzorzec walidacyjny dla numeru telefonu.

Constructor Summary

Constructors

Constructor	Description
Controller(boolean connectDB)	Konstruktor kontrolera.

Ilustracja 2: Fragment dokumentacji JavaDoc klasy Controller z warstwy logiki.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
java.lang.String	addBook(java.lang.String isbn, java.lang.String title, java.lang.String author, java.lang.String publisher)	Metoda dodająca książkę do struktury danych.
java.lang.String	addReader(java.lang.String firstName, java.lang.String lastName, java.time.LocalDate dateOfBirth, java.lang.String phoneNumber, java.lang.String emailAddress)	Metoda dodająca czytelnika do struktury danych.
java.lang.String	addReminder(Rent rent, java.time.LocalDateTime dateOfReminder)	Metoda dodająca przypomnienie o zwrocie książki
java.lang.String	addRent(Book book, Reader reader, java.time.LocalDateTime rentDate, java.lang.String rentalPeriod)	Metoda dodająca wypożyczenie do struktury danych.
private boolean	checkBookIsAvailable(Book book)	Metoda sprawdzająca dostępność książki do wypożyczenia.
private boolean	checkEmailAddress(java.lang.String emailAddress)	Walidacja adresu e-mail
boolean	checkExistsRequiredTables()	Metoda sprawdzająca czy w bazie danych istnieją wymagane tabele.
private boolean	checkIsbn(java.lang.String isbn)	Walidacja numeru ISBN
private boolean	checkPhoneNumber(java.lang.String phoneNumber)	Walidacja numeru kontaktowego
java.lang.String	deleteBook(Book removeBook)	Metoda usuwająca książkę ze struktury danych.
java.lang.String	deleteReader(Reader removeReader)	Metoda usuwająca czytelnika ze struktury danych.
java.lang.String	deleteReminder(RentalReminder rentalReminder)	Metoda usuwająca przypomnienie ze struktury danych.
java.lang.String	exportDataToXML(java.lang.String filePath)	Eksport struktury danych biblioteki do pliku XML.
java.lang.String	extendRentalPeriod(Rent rent, java.lang.String numOfDayExtendRentPeriod)	Metoda do przedłużenia czasu wypożyczenia
java.util.List<Book>	getBooks()	Metoda zwracająca listę książek.
java.util.List<Reader>	getReaders()	Metoda zwracająca listę czytelników.
java.util.List<RentalReminder>	getRentalReminders()	Metoda zwracająca listę przypomnień.
java.util.List<Rent>	getRents()	Metoda zwracająca listę wypożyczeń.
java.lang.String	importDataFromListsToDB()	Import danych z struktury danych biblioteki(listy) do bazy danych.
java.lang.String	importDataFromXML(java.lang.String filePath)	Import danych z pliku XML do struktury danych biblioteki oraz bazy danych.
boolean	isNumeric(java.lang.String string)	Metoda sprawdzająca czy podany ciąg znaków jest liczbą
java.lang.String	returnBook(Rent rent, boolean op)	Metoda zwracająca / cofająca zwrot książki
java.lang.String	updateBook(Book updateBook, java.lang.String isbn, java.lang.String title, java.lang.String author, java.lang.String publisher)	Metoda aktualizująca dane książki w strukturze danych.
java.lang.String	updateReader(Reader updateReader, java.lang.String firstName, java.lang.String lastName, java.time.LocalDate dateOfBirth, java.lang.String phoneNumber, java.lang.String emailAddress)	Metoda aktualizująca dane czytelnika w strukturze danych.
java.lang.String	updateRent(Rent updateRent, java.time.LocalDateTime rentDate, java.lang.String rentalPeriod)	Metoda aktualizująca wypożyczenie w strukturze danych.

Ilustracja 3: Wykaz metod klasy Controller.

Warstwa logiki (kontroler) jest odpowiedzialna za sterowanie aplikacją i komunikację z użytkownikiem poprzez komunikaty przekazywane poprzez warstwę interfejsu. Kontroler pośredniczy tym samym w przetwarzaniu danych pomiędzy modelem, a widokiem aplikacji.

## 5. Najważniejsze klasy warstwy interfejsu

Module `org.pk.library`

Package `org.pk.library.view`

Pakiet z graficznym interfejsem użytkownika wdrożnym w JavaFX oraz terminalowym interfejsem użytkownika.

Class Summary	
Class	Description
<code>App</code>	JavaFX Library App
<code>BookController</code>	Kontroler GUI obsługujący żądania dotyczące książek.
<code>MainController</code>	Główny kontroler GUI.
<code>ReaderController</code>	Kontroler GUI obsługujący żądania dotyczące czytelników.
<code>RentController</code>	Kontroler GUI obsługujący żądania dotyczące wypożyczeń książek.
<code>ReturnController</code>	Kontroler GUI obsługujący żądania dotyczące zwrotów książek.
<code>Settings</code>	Klasa reprezentująca ustawienia aplikacji.
<code>TUI</code>	Kontroler TUI.
<code>TUI.CLS</code>	Klasa wewnętrzna czyszcząca terminal użytkownika.

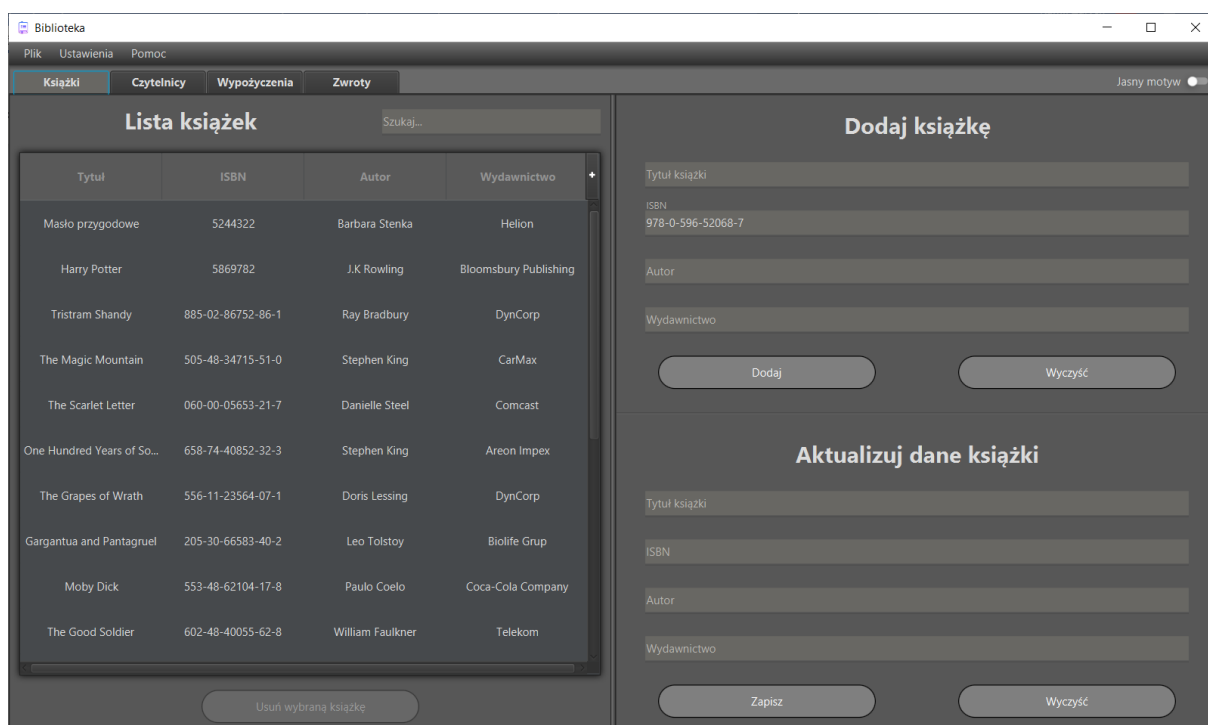
*Ilustracja 4: Pakiet warstwy interfejsu z dokumentacji JavaDoc.*

Warstwa interfejsu (widok) to część aplikacji odpowiedzialna za prezentację danych użytkownikowi. Widoki mogą pobierać odpowiednie dane z modelu, lub modyfikować je poprzez warstwę logiki (kontroler). Warstwa interfejsu graficznego w naszym programie została podzielona na poszczególne kontrolery odpowiedzialne za kontrolowanie odpowiednich części programu.

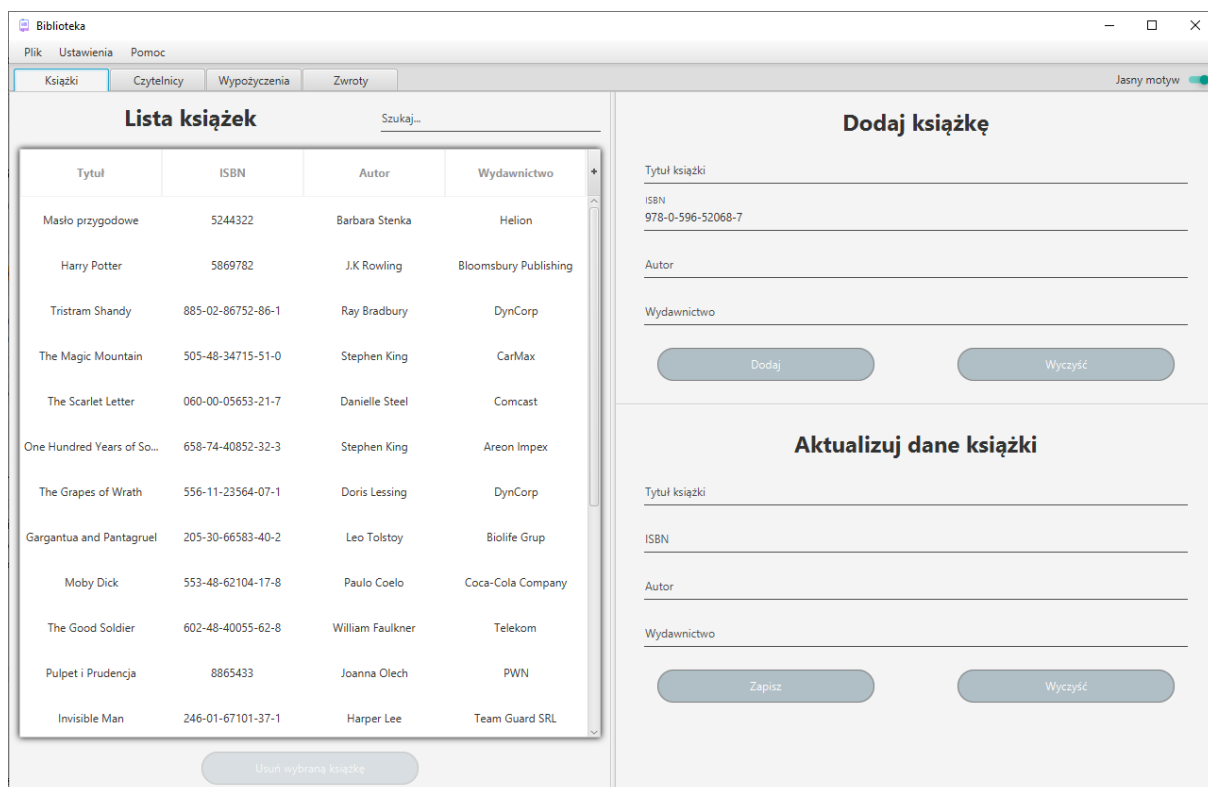
## 6. Zrzuty ekranu najistotniejszych widoków działającej aplikacji

### 6.1 Główne okno programu

Aplikacja posiada cztery główne zakładki – Książki, Czytelnicy, Wypożyczenia, Zwroty. W poszczególnych zakładkach znajdują się tabele zawierające informacje dotyczące danej zakładki oraz pola tekstowe do dodawania / edycji.

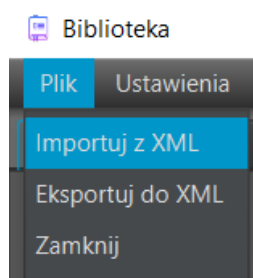


Ilustracja 5: Główne okno aplikacji z domyślnie ustawionym ciemnym motywem.



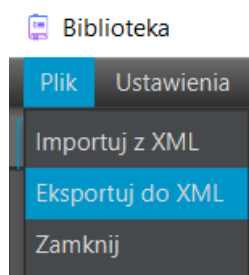
Ilustracja 6: Główne okno aplikacji z jasnym motywem.

**Importuj z XML** – pozwala na import danych z pliku .xml



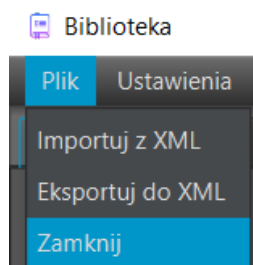
*Ilustracja 7: Menu -> Plik -> Importuj z XML*

**Eksportuj do XML** – pozwala na eksport danych do pliku .xml



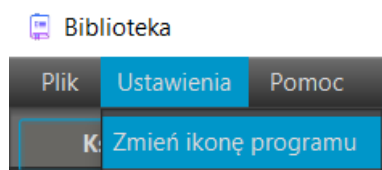
*Ilustracja 8: Menu -> Plik -> Eksportuj do XML*

**Zamknij** – zamyka instancję programu.



*Ilustracja 9: Menu -> Plik -> Eksportuj do XML*

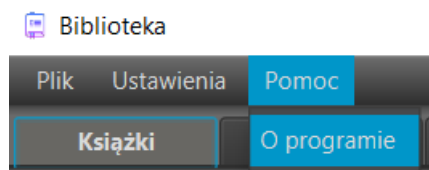
**Zmień ikonę programu** – pozwala na zmianę ikony programu (zmiana widoczna po ponownym uruchomieniu aplikacji).



*Ilustracja 10: Menu -> Ustawienia -> Zmień ikonę programu*



O programie – wyświetla informację o programie.



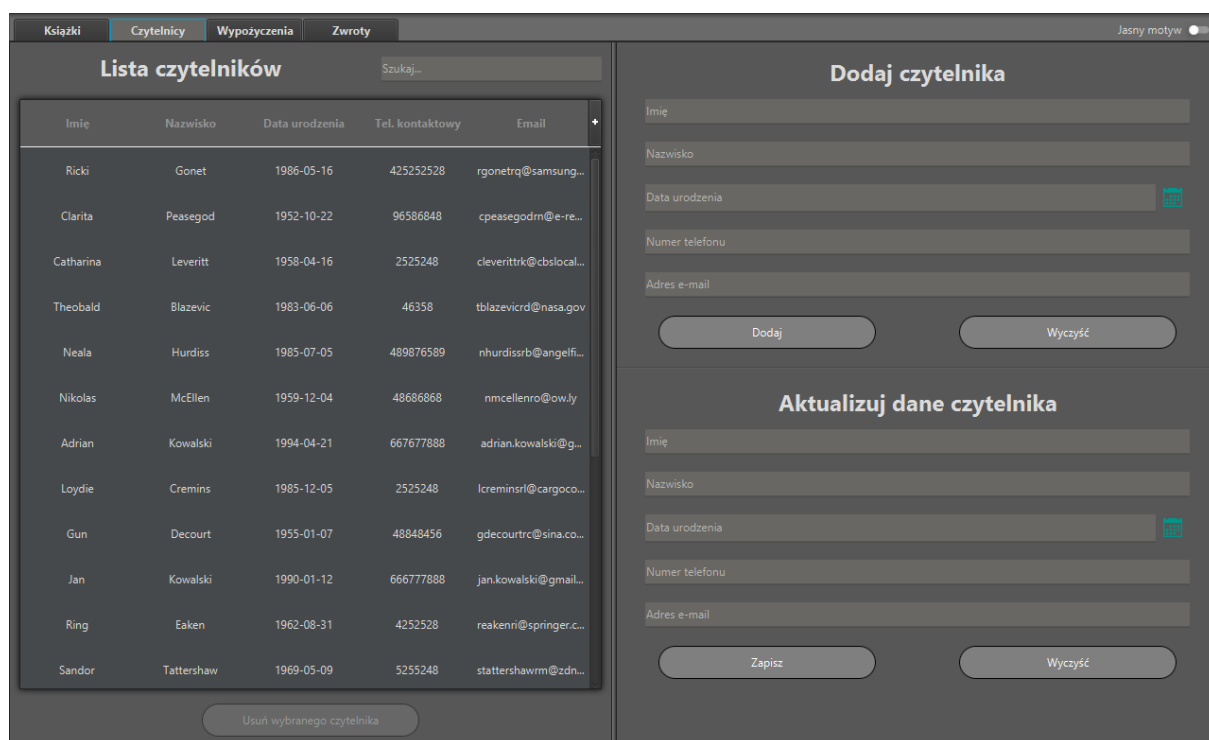
Ilustracja 11: Menu -> Pomoc -> O programie

Przycisk **toggleButton** – pozwala na przełączanie pomiędzy jasnym a ciemnym motywem.



Ilustracja 12: toggleButton – zmiana motywu aplikacji

## 6.2 Zakładka „Czytelnicy” – zarządzanie czytelnikami



Ilustracja 13: Widok zakładki „Czytelnicy”

Widok ten umożliwia wyszukiwanie, filtrowanie, dodawanie i aktualizację danych czytelników. W formularzach zostało zaimplementowane sprawdzanie poprawności wprowadzanych danych tj. puste pola, numeru telefonu i adres e-mail.

### 6.3 Zakładka „Wypożyczenia” – zarządzanie wypożyczeniami

**Lista wypożyczeń**

Książka	Czytelnik	Data wypożyczenia	Data zwrotu	Czy zwrócono
The Scarlet Letter	Ky Dumingo	2020-06-07T17:40	2020-06-26T17:40	nie
Harry Potter	Jan Kowalski	2020-06-07T00:00	2020-05-10T02:30:01	nie
The Flowers of Evil	Jan Kowalski	2020-06-07T17:40	2020-06-19T17:40	nie
One Hundred Years...	Darb Shorto	2020-06-07T18:40	2020-06-22T18:40	nie
Great Expectations	Beatrix Sorro	2020-06-07T17:40	2020-06-28T17:40	nie
The Magic Mountain	Ky Dumingo	2020-06-07T17:40	2020-06-28T17:40	nie
Moby Dick	Tamra Chatenet	2020-06-07T18:40	2020-06-16T18:40	tak

**Wypożyczanie książki**

Wybierz czytelnika: Szukaj...

Imię	Nazwisko	Data urodzenia
Ricki	Gonet	1986-05-16
Clarita	Peasegod	1952-10-22
Catharina	Leveritt	1958-04-16
Theobald	Blazevic	1983-06-06
Neala	Hurdiss	1985-07-05
Nikolas	McEllen	1959-12-04
Adrian	Kowalski	1994-04-21
Loydie	Cremins	1985-12-05

Wybierz książkę: Szukaj...

Tytuł	Autor
Masło przygodowe	Barbara Stenka
Harry Potter	J.K. Rowling
Tristram Shandy	Ray Bradbury
The Magic Mountain	Stephen King
The Scarlet Letter	Danielle Steel
One Hundred Years of ...	Stephen King
The Grapes of Wrath	Doris Lessing
Gargantua and Pantagr...	Leo Tolstoy

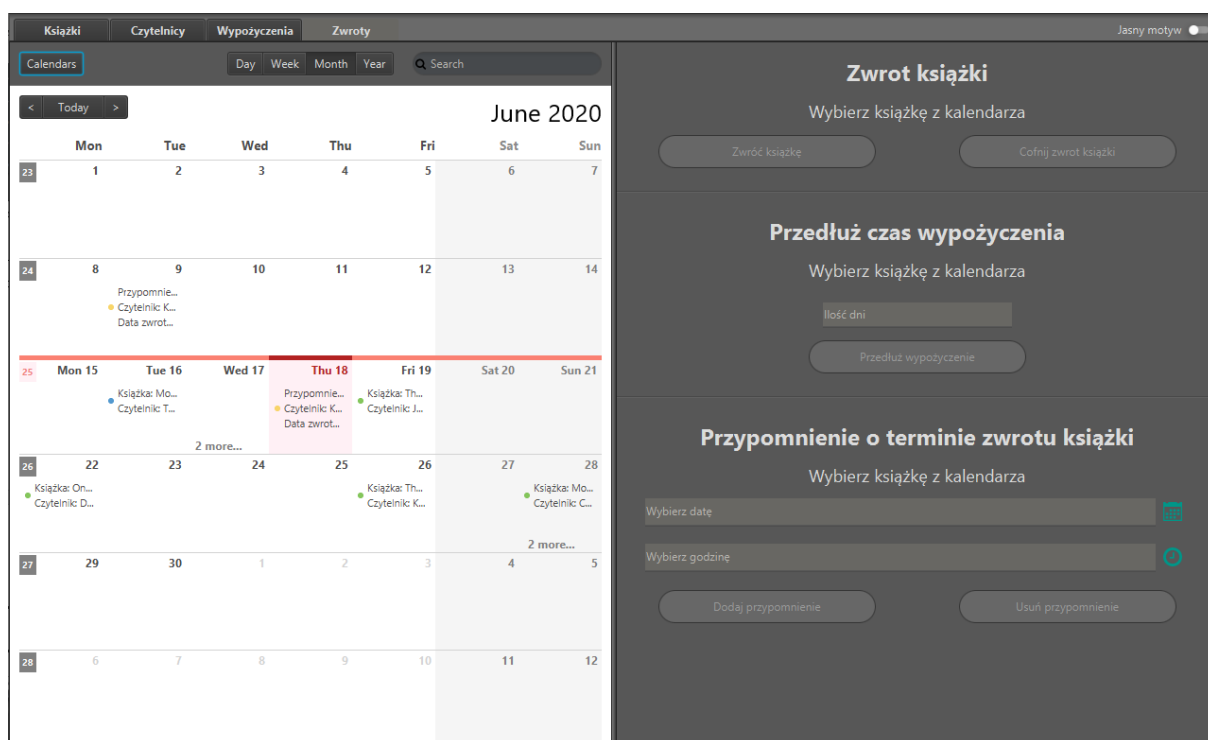
Data wypożyczenia: Godzina wypożyczenia: Ilość dni:

**Dodaj wypożyczenie** **Aktualizuj wypożyczenie**

Ilustracja 14: Widok zakładki „Wypożyczenia”

Widok ten umożliwia wyszukiwanie, filtrowanie, dodawanie i aktualizację danych wypożyczeń książek. Aby wypożyczyć książkę należy wybrać czytelnika i książkę, datę i godzinę wypożyczenia oraz ilość dni, na którą książka będzie wypożyczona. W przypadku pozostawienia pól daty i godziny pustych zostaną wczytane aktualne dane z systemu. Data wypożyczenia może być podana maksymalnie 7 dni wstecz. W przypadku braku dostępności książki wyświetlony zostaje odpowiedni komunikat.

## 6.4 Zakładka „Zwroty” – zarządzanie zwrotami książek i przypomnieniami



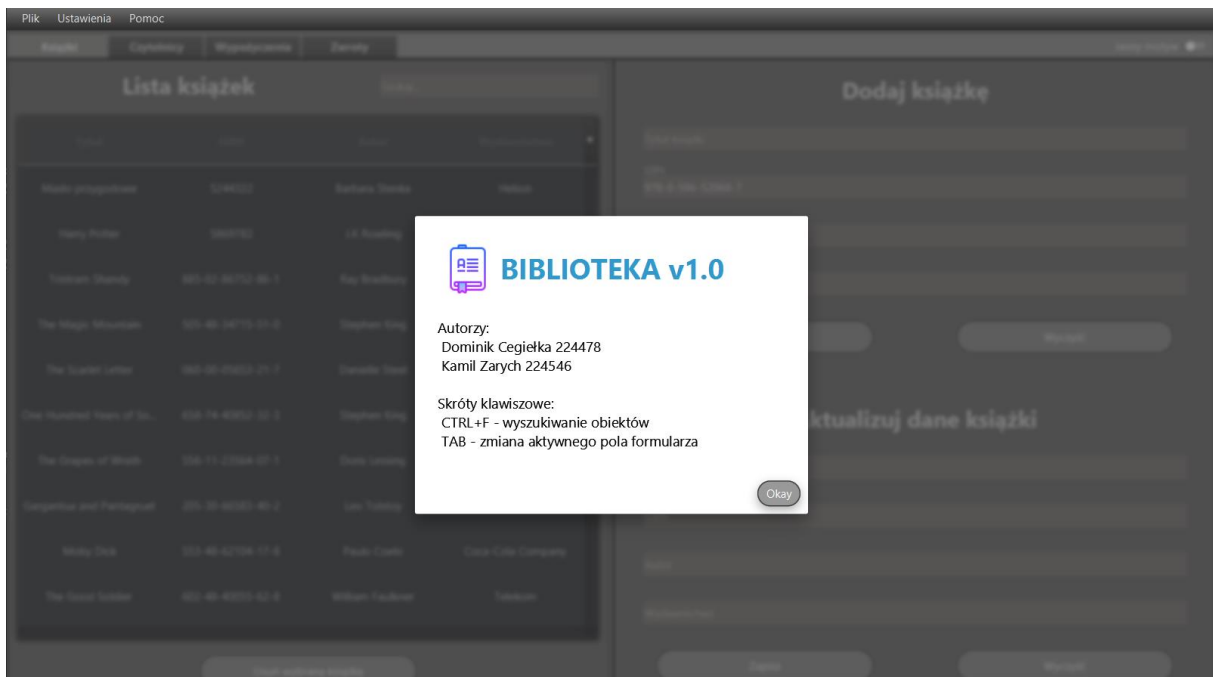
Ilustracja 14: Widok zakładki „Zwroty

Widok ten umożliwia zwrot wypożyczonych książek, przedłużenie czasu wypożyczenia wybranej książki oraz dodawanie do kalendarza przypomnień o terminie zwrotu książki. Widok kalendarza zawiera trzy kalendarze: wypożyczenia aktualne, wypożyczenia archiwalne oraz przypomnienia. Ponadto widok ten umożliwia wyszukiwanie wypożyczeń i powiadomień po dowolnych danych.

Przypomnienia możemy przeglądać w kalendarzu i są one również wyświetlane jako powiadomienia dialogowe w chwili wystąpienia przypomnienia.

Zwrot książki jest możliwy po wybraniu wypożyczenia z kalendarza wypożyczeń aktualnych a wycofanie zwrotu książki po wybraniu wypożyczenia z kalendarza wypożyczeń archiwalnych.

### 6.5 Okienko „O programie” z opisem klawiszy funkcyjnych

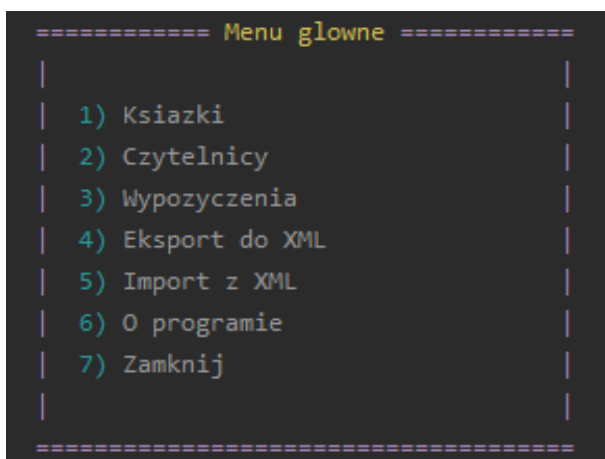


Ilustracja 13: Okienko „O programie”.

## 6.6 Dodatkowe funkcjonalności

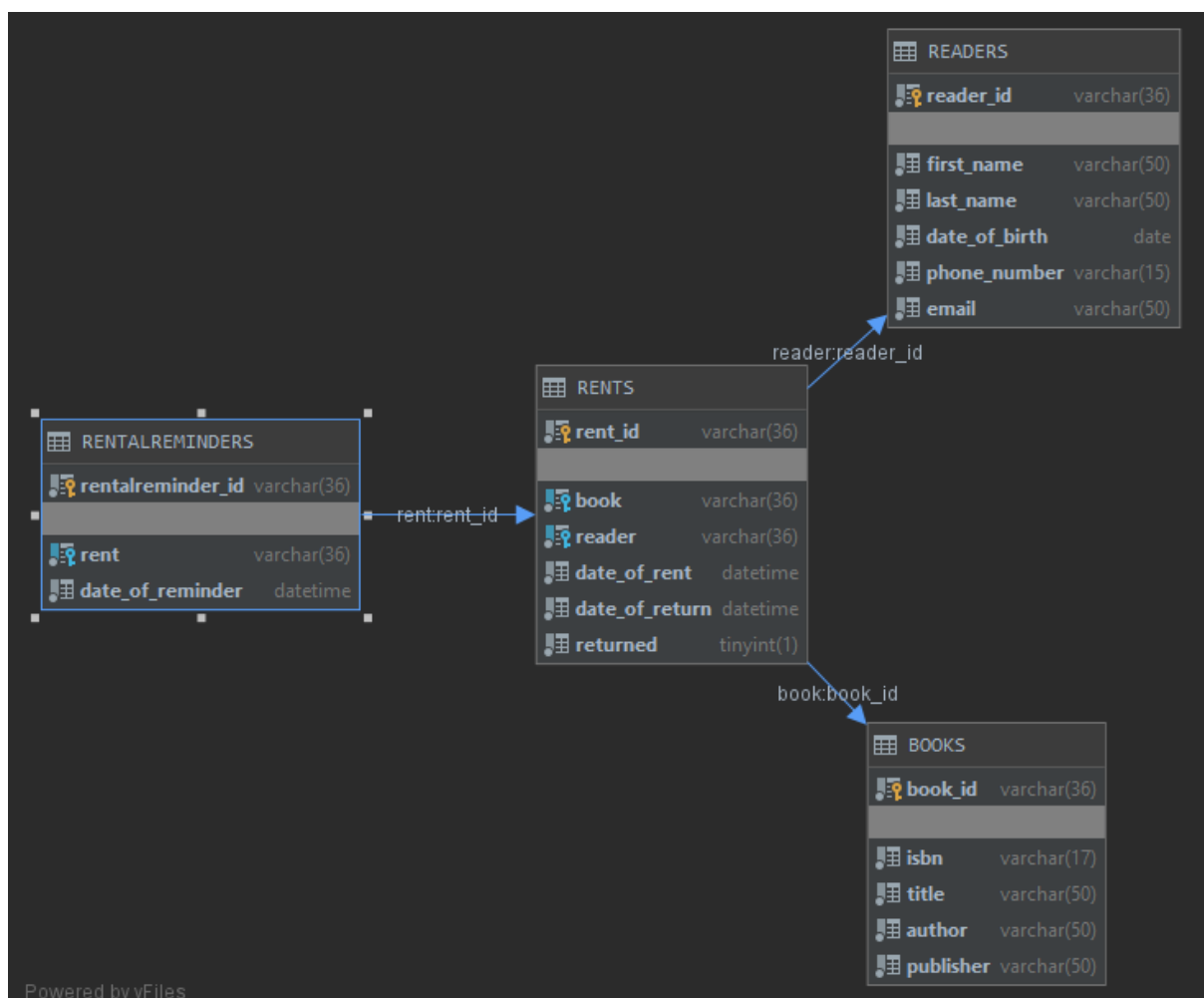
Interfejs graficzny jest responsywny. Zaimplementowaliśmy również interfejs konsolowy, który obsługuje:

- Dodanie / usunięcie książki,
- Wyświetlenie listy książek,
- Dodanie / usunięcie czytelnika,
- Wyświetlenie listy czytelników,
- Dodanie wypożyczenia,
- Wyświetlenie listy wypożyczeń,
- Wyświetlenie listy przypomnień,
- Import / Eksport z / do pliku XML,
- „O programie”



*Ilustracja 14: Menu główne w TUI*

## 7. Postać bazy danych



Ilustracja 15: Schemat bazy danych biblioteki

	isbn	title	author	publisher	book_id
1	5244322	Masio przygodowe	Barbara Stenka	Helion	0f571dd0-3d4c-4d93-9e03-2285c7036642
2	5869782	Harry Potter	J.K Rowling	Bloomsbury Publishing	0f789467-92b9-4c58-80f0-856338ad8230
3	885-02-86752-86-1	Tristram Shandy	Ray Bradbury	DynCorp	20c85d48-dd48-4c25-8b03-b3d5af722741
4	505-48-34715-51-0	The Magic Mountain	Stephen King	CarMax	2c4f04d1-ceb6-4628-8bf1-c56b8da2df11
5	060-00-05653-21-7	The Scarlet Letter	Danielle Steel	Comcast	33e85c59-b84e-4582-a6ed-9257f90f9069

Ilustracja 16: Tabela danych książek

	first_name	last_name	date_of_birth	phone_number	email	reader_id
1	Ricki	Gonet	1986-05-16	425252528	rgonetrg@samsung.com	03ed074d-ea0b-4389-bee4-fb749651c613
2	Clarita	Peasegod	1952-10-22	96586848	cpeasegodrn@recht24.de	1f7113fa-0656-4298-b5a2-112348a2d3e4
3	Catharina	Leveritt	1958-04-16	2525248	cleverittrk@cbslocal.com	24f31d2c-bfb8-4b56-b8bf-b86e84f4dd87
4	Theobald	Blazevic	1983-06-06	46358	tblazevicrd@nasa.gov	3e343110-dc23-4bd0-804c-63cf54455395
5	Neala	Hurdiss	1985-07-05	489876589	nhurdissrb@angelfire.com	46fa4a95-23e8-414c-91a5-ff6dbf0dfae4

Ilustracja 17: Tabela danych czytelników

	book	reader	date_of_rent	date_of_return	returned	rent_id
1	33e85c59-b84e-4582-a6ed-9257f90f9069	ec281180-a19d-4a5b-8122-39ba752157ff	2020-06-07 17:40:00	2020-06-26 17:40:00	0	07ad5432-5751-4122-9de0-069a3cbd1305
2	0f789467-92b9-4c58-80f0-856338ad8230	7da9d877-3530-46e8-a2b9-f4731fff5726	2020-06-07 00:00:00	2020-05-10 02:30:01	0	7da9d877-3530-46e8-a2b9-f4731fff5726
3	ac22b306-9343-4d6b-9119-49c18755822c	7da9d877-3530-46e8-a2b9-f4731fff5726	2020-06-07 17:40:00	2020-06-19 17:40:00	0	8442e95e-cdc6-47a3-b81d-da9214136f52
4	3c17461b-72ad-412e-82f6-ab04b713775d	cbabf867-c253-43dc-b714-685b6582b216	2020-06-07 18:40:00	2020-06-22 18:40:00	0	9b235122-7096-4269-8513-17fa717699a1
5	e4042e16-6c31-4022-9bda-aae78f48c7ba	dac818a3-a651-4e10-89d4-f5342f520739	2020-06-07 17:40:00	2020-06-28 17:40:00	0	b657336a-1891-4299-8a11-6bf0d411c306

Ilustracja 18: Tabela danych wypożyczeń

	rent	date_of_reminder	rentalreminder_id
1	9b235122-7096-4269-8513-17fa717699a1	2020-06-17 15:49:00	47560253-ae2f-4765-9fc5-50dad16691d3
2	b657336a-1891-4299-8a11-6bf0d411c306	2020-06-17 12:49:00	56e650e7-81cb-444e-b3ac-7f56d7477d3f
3	7da9d877-3530-46e8-a2b9-f4731fff5726	2020-05-08 02:30:01	75daf846-c56b-4c8b-8073-074ee34d9004
4	e1b94734-ff64-40df-b55f-85205fdc45ff	2020-06-09 13:49:00	aa21bc9c-fa44-456f-b2c1-fe3fcb8a5b80
5	e1b94734-ff64-40df-b55f-85205fdc45ff	2020-06-18 14:50:00	cc84b230-fd7e-4340-9514-5b20d50358e5

*Ilustracja 19: Tabela danych przypomnień*

## 8. Podsumowanie

Zaimplementowany przez nas program posiada wszystkie wymagane funkcjonalności, a ponadto posiada responsywny interfejs graficzny użytkownika. Implementacja dwóch interfejsów użytkownika (graficzny i terminalowy) była ułatwiona dzięki zastosowaniu modelu MVC i wykorzystaniu jednej warstwy logiki a w konsekwencji jednej warstwy dostępu do danych. Zastosowanie tego typu podejścia upraszcza proces tworzenia kolejnych interfejsów użytkownika oraz dalszej rozbudowy programu o kolejne funkcjonalności w przypadku zaistnienia takiej potrzeby w przyszłości.