****

**Praca projektowa programowanie obiektowe**

*Program wspierający biblioteki w procesie obsługi czytelników*

Rzeszów 2024

Spis treści

[**1.** **Opis założeń projektu** 3](#_Toc157246319)

[**2.** **Specyfikacja wymagań** 5](#_Toc157246320)

[**2.1.** **Wymagania funkcjonalne** 5](#_Toc157246321)

[**2.2.** **Wymagania niefunkcjonalne** 5](#_Toc157246322)

[**3.** **Opis struktury projektu** 7](#_Toc157246323)

[**4.** **Harmonogram realizacji projektu** 11](#_Toc157246324)

[**5.** **Prezentacja warstwy użytkowej projektu** 12](#_Toc157246325)

[**6.** **Podsumowanie** 19](#_Toc157246326)

[**6.1.** **Zrealizowane prace** 19](#_Toc157246327)

[**6.2.** **Możliwe dalsze prace rozwojowe** 19](#_Toc157246328)

[**7.** **Literatura** 21](#_Toc157246329)

1. **Opis założeń projektu**

Codziennie pracownicy bibliotek stają przed wyzwaniem zapewnienia sprawnej i efektywnej obsługi czytelników, jednocześnie zarządzając rozległą bazą książek i informacji o użytkownikach. Podobnie jak w biurze, gdzie drobne niedociągnięcia w codziennych obowiązkach mogą zaburzać harmonię pracy, tak w bibliotece zaniedbania w obszarze zarządzania księgozbiorami i czytelnikami mogą prowadzić do frustracji zarówno pracowników, jak i użytkowników. Typowe problemy, takie jak trudności w odnalezieniu konkretnej książki, opóźnienia w procesie wypożyczania, czy brak aktualnych danych o czytelnikach, mogą znacznie obniżać efektywność pracy i zadowolenie czytelników. Aby sprostać tym wyzwaniom, nasz zespół opracował rozwiązanie – aplikację desktopową dla pracowników bibliotek, która stanowi odpowiednik dobrze zorganizowanego i efektywnego biurowego środowiska pracy. Nasz system wspierający biblioteki w procesie obsługi czytelników jest zaprojektowany tak, aby ułatwić codzienne zadania związane z zarządzaniem księgozbiorem, jak i relacjami z czytelnikami.

**Cele i założenia projektu:**

**Cel projektu:** Celem projektu jest stworzenie aplikacji desktopowej dla pracowników bibliotek, która ułatwi i zautomatyzuje procesy zarządzania księgozbiorem oraz obsługą czytelników. System ma za zadanie nie tylko poprawić efektywność pracy bibliotekarzy, ale również zwiększyć satysfakcję czytelników poprzez szybsze i dokładniejsze usługi związane z wypożyczaniem i zwracaniem tytułów.

**Problem i jego źródło:** Głównym problemem, który zostanie rozwiązany, jest niewystarczająca efektywność i organizacja w procesach zarządzania księgozbiorem oraz obsługi czytelników w bibliotekach. Podstawowym źródłem problemu jest brak zintegrowanych, łatwych w obsłudze narzędzi informatycznych, które by wspierały te procesy, co prowadzi do czasochłonnego wyszukiwania książek, błędów w rejestracji wypożyczeń oraz trudności w zarządzaniu danymi czytelników.

**Ważność problemu i dowody:** Problem ten jest istotny, ponieważ wpływa na jakość usług świadczonych przez biblioteki, a także na zadowolenie zarówno pracowników, jak i czytelników. Dowody na istnienie tego problemu znajdujemy w opóźnieniach w procesie wyszukiwania i wypożyczania książek, skargach użytkowników oraz w obserwacji niskiej efektywności pracy bibliotekarzy przy użyciu przestarzałych systemów.

**Co jest niezbędne do rozwiązania problemu:** Aby rozwiązać ten problem, niezbędne jest opracowanie nowoczesnego oprogramowania, które umożliwi integrację danych księgozbioru i czytelników, automatyzację procesów wypożyczania i zwrotu książek oraz łatwe zarządzanie danymi. Kluczowe będzie zapewnienie intuicyjnego interfejsu użytkownika oraz funkcji eksportowania i importowania danych w formacie CSV.

**Sposób rozwiązania problemu:** Problem zostanie rozwiązany poprzez etapowy rozwój aplikacji. Rozpoczniemy od analizy potrzeb i wymagań użytkowników, następnie przejść do projektowania interfejsu użytkownika i architektury systemu. Kolejnym krokiem będzie programowanie funkcjonalności takich jak wyszukiwanie książek, rejestracja wypożyczeń i zwrotów, zarządzanie danymi czytelników, oraz eksport/import danych. Po wdrożeniu systemu nastąpi faza testów i ewentualnych poprawek. Wynikiem projektu będzie funkcjonalna i użyteczna aplikacja, która znacząco usprawni pracę bibliotekarzy i poprawi jakość obsługi czytelników.

Założeniem projektu „Biblioteka” było stworzenie programu wspierającego pracowników biblioteki w procesie obsługi czytelników, zapewniając możliwość zarządzania tytułami i czytelnikami. Pracownik biblioteki po uprzednim zalogowaniu się do systemu ma przede wszystkim możliwość dodawania i usuwania czytelników oraz dodawania książek, wypożyczania ich i zwracania.

Każdy czytelnik jest reprezentowany przez niepowtarzalny, indywidualny ID czytelnika, imię i nazwisko oraz nr PESEL. Książki są reprezentowane swoim tytułem, autorem oraz liczbą dostępnych egzemplarzy. Natomiast wypożyczenia są reprezentowane za pomocą ID czytelnika który je wypożyczył, tytułem i autorem książki oraz datą wypożyczenia, która jest później wykorzystywana przy zwrotach książek do obliczania kary za opóźnienia. Pracownik biblioteki ma również możliwość wyszukiwania konkretnych tytułów, aby sprawdzić ich dostępność.

Do dyspozycji pracownika jest również opcja eksportowania bazy książek i czytelników do pliku CSV jak i również ich importowania z plików CSV.

Po każdej wykonanej operacji wyświetlane są odpowiednie komunikaty informujące o powodzeniu bądź wystąpieniu błędu podczas wykonywania zadania. Zakłada się również, iż pliki bazy danych nie będą przechowywać rekordów w liczbach większych niż 150.

Zaleca się język Java + Swing + Maven + bibliotekę OpenCSV

1. **Specyfikacja wymagań**

## **Wymagania funkcjonalne**

* Autoryzacja pracownika biblioteki przebiega na podstawie wprowadzonego loginu i hasła w panelu logowania, za pomocą danych przekazanych pracownikowi przez administratora systemu.
* Pracownicy biblioteki mają dostęp do systemu po uprzednim zalogowaniu się. W przypadku pomyślnego zalogowania się, otrzymują dostęp do danych czytelników, informacji o wypożyczeniach oraz liście książek wraz z liczbą dostępnych w danej chwili egzemplarzy
* Program oferuje pracownikom wsparcie w procesie obsługi czytelników, umożliwiając im w szybki sposób sprawdzanie dostępności tytułów (czy biblioteka posiada dany tytuł oraz czy jest on możliwy do wypożyczenia) zarówno w formie tabeli na głównym panelu programu, jak po wybraniu funkcji wyszukiwania danego tytułu w bazie danych. Program zapewnia również możliwość szybkiego i łatwego wypożyczania i zwracania książek przez czytelników
* Pracownik oprócz przeszukiwania bazy książek w celu sprawdzenia dostępności danego tytułu, może również dodawać do niej nowe utwory, które natychmiast będą możliwe do wypożyczenia przez czytelnika. Oprócz książek, program oczywiście oferuje również dodawanie czytelnika do bazy
* Program oferuje czytelnikowi, za pośrednictwem pracownika biblioteki oraz samemu pracownikowi, usuwanie danych z bazy czytelników.
* Program oferuje również pracownikom import i eksport danych z/do pliku CSV, co umożliwia łatwe przywracanie i przenoszenie danych.
* Zakończenie wykonywanych operacji jest oznajmiane użytkownikowi stosownym komunikatem informującym o powodzeniu bądź niepowodzeniu w wykonywaniu wybranej funkcji programu

## **Wymagania niefunkcjonalne**

* Aplikacja jest przyjazna pracownikom i działa na wielu urządzeniach, dzięki stosowaniu Java bytecode.
* Aplikacja tworzona jest w języku JAVA z zastosowaniem biblioteki Swing dla graficznego interfejsu użytkownika, narzędzia Maven oraz biblioteki OpenCSV.
* Aplikacja powinna szybko reagować na zapytania użytkowników, zapewniając krótkie czasy odpowiedzi.
* Optymalizacja interakcji z lokalną bazą danych w plikach tekstowych, aby zapewnić płynne działanie aplikacji nawet przy dużym obciążeniu.
* Prosty, intuicyjny, łatwy w obsłudze interfejs użytkownika, zaprojektowany z myślą o użytkownikach o różnym stopniu zaawansowania technologicznego.
* Możliwość łatwego rozszerzania aplikacji o nowe funkcje.
* Zdolność do obsługi rosnącej liczby książek i czytelników bez utraty wydajności.
* Zapewnienie niezawodnego działania na wspieranych systemach operacyjnych
* Zapewnienie zgodności z lokalnymi i międzynarodowymi przepisami dotyczącymi ochrony danych osobowych (np. RODO/GDPR).
* Łatwość w utrzymaniu i aktualizowaniu aplikacji, w tym regularne łatki bezpieczeństwa i aktualizacje funkcjonalne.
* Optymalizacja kosztów związanych z rozwojem, utrzymaniem i eksploatacją aplikacji.

1. **Opis struktury projektu**

**Zastosowany język, środowisko programistyczne i użyte narzędzia:**

**Java JDK:** Zalecana wersja Oracle OpenJDK version 19.0.2 – najnowsze wydanie otwarto źródłowej implementacji Java Development Kit (JDK) od Oracle, zapewniającej zaawansowane środowisko do tworzenia aplikacji Java. Ta wersja zawiera aktualizacje zabezpieczeń, ulepszenia wydajności oraz nowe funkcje, zgodnie z regularnym harmonogramem wydawania JDK, co zapewnia deweloperom aktualne i stabilne narzędzia do programowania.

**IDE**: IntelliJ IDEA w wersji 2023.1.5 (Community Edition) – zaawansowane zintegrowane środowisko programistyczne (IDE) firmy JetBrains, zaprojektowane głównie dla języka Java. Wersja ta przynosi szereg ulepszeń i poprawek, zwiększając wydajność, ułatwiając nawigację w kodzie oraz usprawniając integrację z narzędziami i frameworkami, co czyni ją jeszcze bardziej efektywną i przyjazną dla programistów.

**GUI Toolkit:** Swing - użyta w projekcie wersja Swing jest zintegrowana z JDK, dlatego zaleca się użycie tej samej wersji JDK dla kompatybilności. Swing jest biblioteką Java służącą do kreowania graficznego interfejsu użytkownika (GUI) dla aplikacji. Zapewnia szeroką gamę komponentów, w tym okna dialogowe, przyciski, pola tekstowe, listy rozwijane, które można swobodnie modyfikować i stylizować. W projekcie, Swing jest wykorzystywany do zbudowania responsywnego i intuicyjnego interfejsu, ułatwiając pracownikom biblioteki płynną i efektywną pracę z aplikacją.

**Maven**: Narzędzie to zostało zastosowane do obsługi pakietu OpenCSV w projekcie - jest to narzędzie do automatyzacji budowy oprogramowania, używane głównie w projektach Java, które pomaga w zarządzaniu całym cyklem życia budowy oprogramowania, od kompilacji, przez testowanie, aż po dystrybucję.

**OpenCSV**: Zastosowana wersja OpenCSV 5.4 - to biblioteka Java przeznaczona do łatwego i wydajnego odczytu oraz zapisu danych w formacie CSV (Comma-Separated Values). Oferuje ona szereg funkcji, takich jak bindowanie danych z/do obiektów Java, obsługę niestandardowych separatorów i znaków otaczających tekst, co czyni ją niezwykle przydatną w aplikacjach wymagających przetwarzania danych tabelarycznych.

**Minimalne wymagania sprzętowe:**

* **Procesor:** Intel Core i3 lub równoważny AMD (minimalnie 2 rdzenie, 2.0 GHz lub wyższy).
* **Pamięć RAM:** Minimum 4 GB RAM, zalecane 8 GB dla lepszej wydajności.
* **Przestrzeń Dyskowa:** Minimum 250 MB wolnej przestrzeni na dysku twardym dla samej aplikacji i dodatkowe miejsce na dane użytkownika oraz bazy danych.
* **System Operacyjny:** Windows 10 lub nowszy z zainstalowaną maszyną wirtualną Javy

Ze względów na liczbę szczegółów aplikacja była tworzona na komputerze z systemem Windows 11

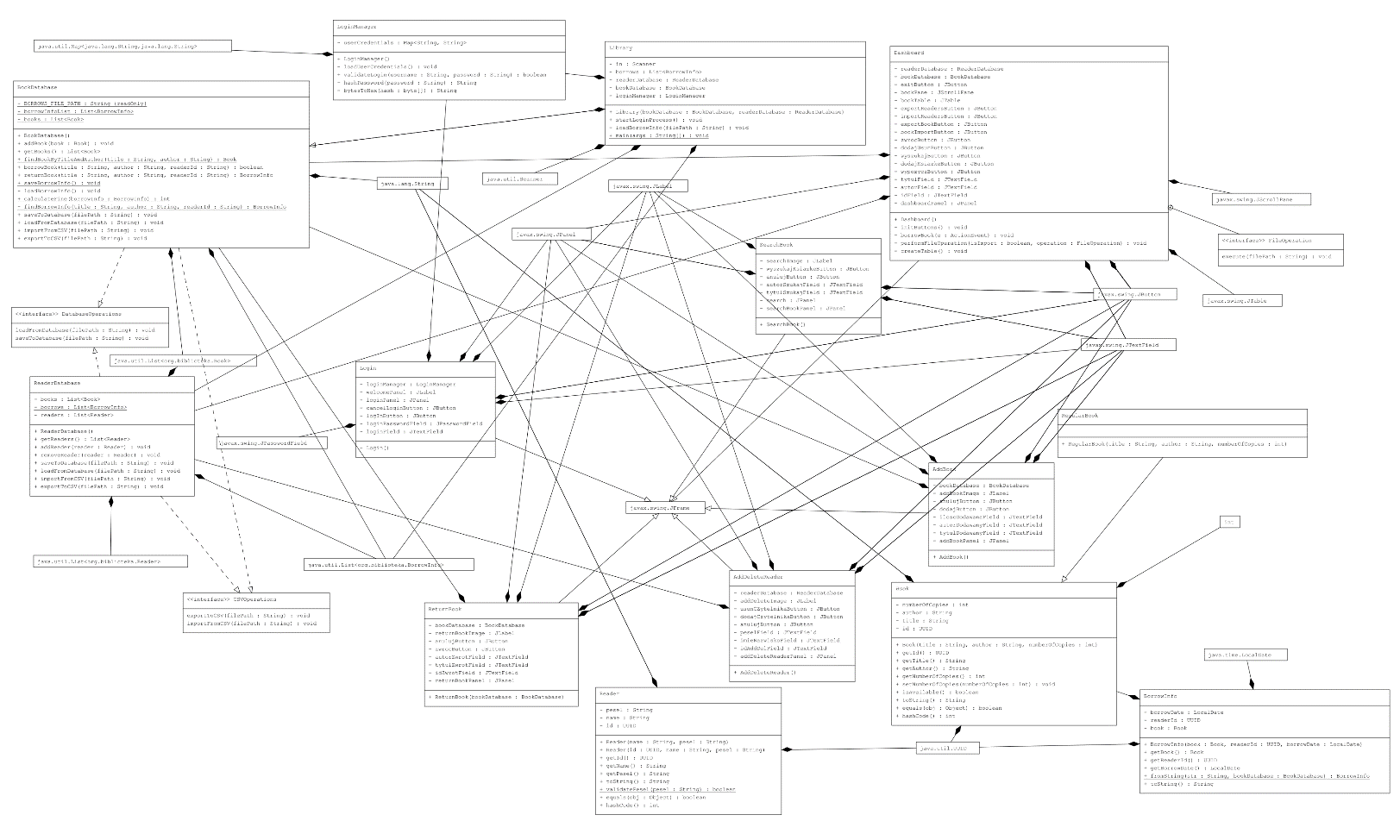


Diagram klas projektowanej aplikacji

Struktura przedstawiona na diagramie klas odnosi się do architektury oprogramowania wspierającego biblioteki w procesie obsługi czytelników. Zawiera on klasy i okna reprezentujące różne aspekty systemu:

1. **Library**: Główna klasa inicjująca działanie aplikacji i uruchamiająca panel logowania użytkownika-pracownika.
2. **LoginManager**: Klasa odpowiadająca za obsługę logowania się użytkownika do systemu.
3. **BookDatabase**: Klasa odpowiedzialna za wczytanie do programu bazy książek oraz wypożyczeń, a także za obsługę wszystkich operacji wykonywanych na plikach bazy danych związanych z książkami i wypożyczeniami. Zapewnia również obsługę importu i eksportu książek z/do pliku CSV.
4. **ReaderDatabase**: Klasa odpowiedzialna za wczytanie do programu bazy czytelników, a także za obsługę wszystkich operacji wykonywanych na plikach bazy danych związanych z czytelnikami. Zapewnia również obsługę importu i eksportu danych czytelników z/do pliku CSV.
5. **BorrowInfo**: Klasa przechowuje informacje o wypożyczeniach, w tym dane książki, czytelnika oraz daty wypożyczenia i zwrotu.
6. **CSVOperations**: Interfejs definiuje metody do obsługi operacji na plikach CSV, takich jak eksport i import danych.
7. **DatabaseOperations**: Interfejs określa ogólne metody do zarządzania bazą danych, takie jak dodawanie, usuwanie i wyszukiwanie danych.
8. **Reader**: Klasa reprezentuje czytelnika, przechowując jego dane osobowe i historię wypożyczeń.
9. **RegularBook:** Klasa rozszerza klasę „Book”, reprezentując standardową książkę z określonymi atrybutami i zachowaniami.
10. **Book**: Jest klasą bazową dla książek, przechowując informacje, takie jak tytuł, autor i liczbę egzemplarzy danego tytułu.
11. **Login**: Okno wyświetlające i obsługujące (zdarzenia, komunikaty) panel logowania użytkownika-pracownika.
12. **Dashboard**: Okno widoczne po pomyślnym zalogowaniu się. Główny panel całej aplikacji zapewniający obsługę wypożyczeń z wykorzystaniem ww. klas. Zawiera przyciski-odnośniki do pozostałych okien obsługujących funkcje aplikacji.
13. **AddBook**: Okno widoczne po wybraniu odpowiedniej opcji na panelu Dashboard. Zapewnia obsługę dodawania książki z wykorzystaniem ww. klas.
14. **ReturnBook**: Okno widoczne po wybraniu odpowiedniej opcji na panelu Dashboard. Zapewnia obsługę zwracania książki wypożyczonej przez danego czytelnika identyfikowanego po unikalnym identyfikatorze, z wykorzystaniem ww. klas.
15. **SearchBook**: Okno widoczne po wybraniu odpowiedniej opcji na panelu Dashboard. Zapewnia obsługę wyszukiwania książki w bazie danych, w celu sprawdzenia jej istnienia w zasobach biblioteki oraz ewentualnej dostępności egzemplarzy, z wykorzystaniem ww. klas.
16. **AddDeleteReader**: Okno widoczne po wybraniu odpowiedniej opcji na panelu Dashboard. Zapewnia obsługę dodawania i usuwania czytelnika z bazy danych czytelników, z wykorzystaniem ww. klas.

Każda klasa zawiera metody i atrybuty odpowiadające za konkretne funkcje programu. Na przykład, w klasie ‘BookDatabase’ znajdują się metody ‘addBook’, która dodaje książkę do bazy danych książek oraz ‘borrowBook’, która realizuje proces wypożyczenia książki przez czytelnika. Innym przykładem jest klasa ‘LoginManager’ w której znajduje się metoda ‘validateLogin’, sprawdzająca poprawność wpisanych danych logowania użytkownika z tymi wprowadzonymi przez administratora do bazy pracowników. Kolejnym przykładem jest klasa ‘ReaderDatabase’, w której znajdują się takie metody jak np.: ‘addReader’ dodająca czytelników do bazy danych oraz metoda ‘exportToCSV, która eksportuje dane czytelników do pliku CSV.

Każda z metod odzwierciedla fundamentalne operacje związane z zarządzaniem księgozbiorem, obsługą wypożyczeń i zarządzaniem danymi czytelników.

1. **Harmonogram realizacji projektu**

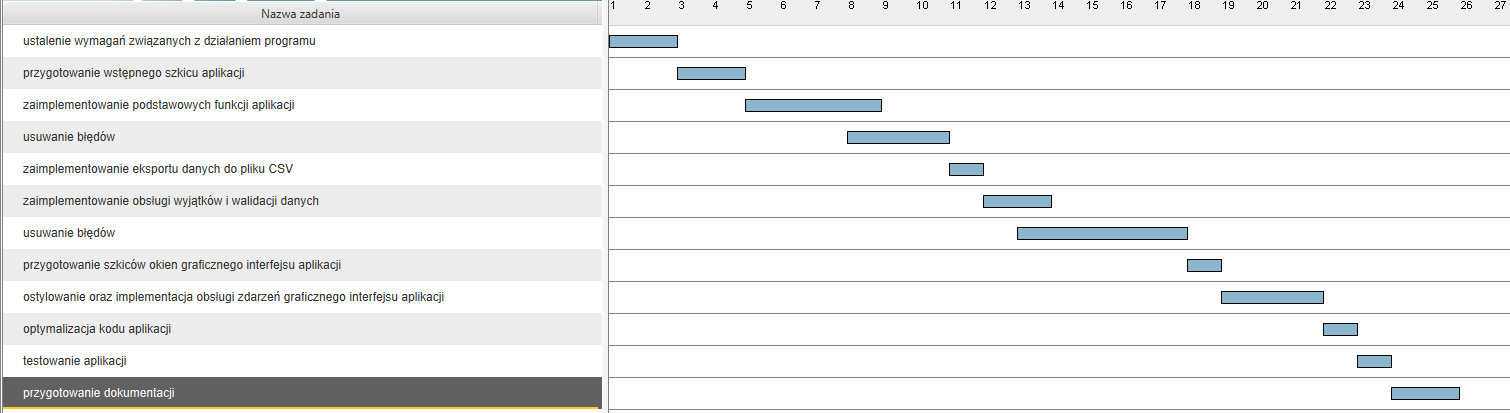
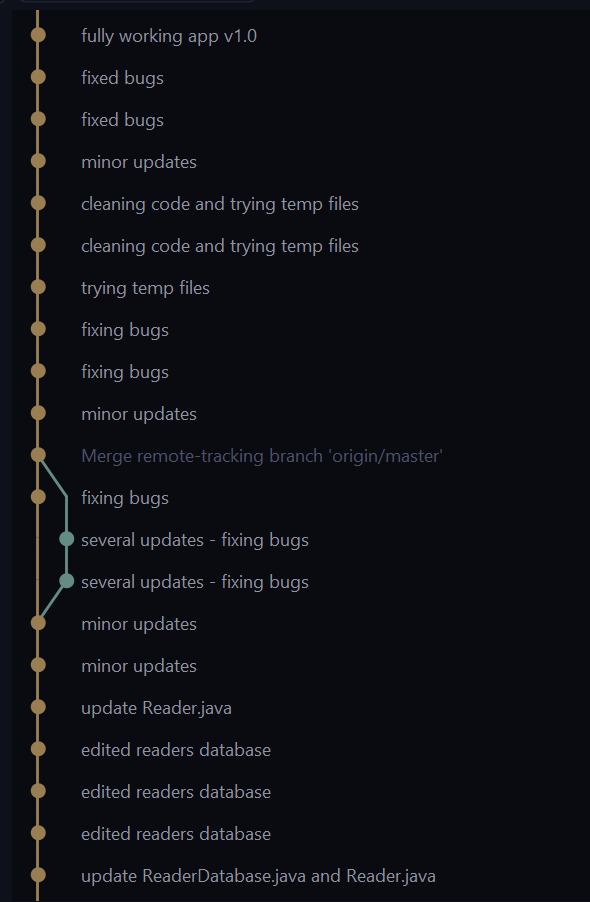
Poniżej zamieszczono harmonogram realizacji projektu

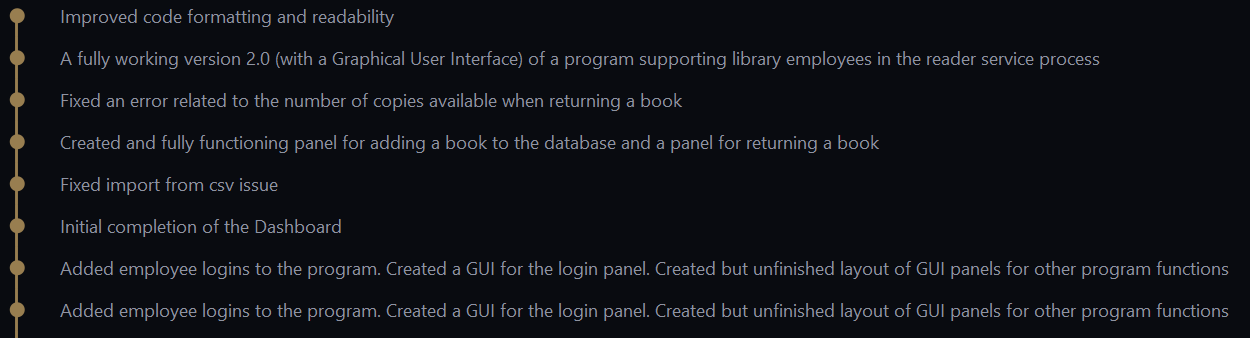
Diagram Gantta

Przedstawiony powyżej diagram Gantta przedstawia harmonogram prac podczas realizacji projektu, zaznaczając najważniejsze etapy tworzenia aplikacji wraz z jednostkami czasu przeznaczonymi na ich realizację. Diagram przedstawia procesy oraz poświęcony czas na zrealizowania stworzenia części logicznej aplikacji, wraz z usuwaniem napotkanych błędów w kodzie, przygotowania szkiców okien graficznego interfejsu użytkownika aplikacji, a następnie ich dokładnego ostylowania. Diagram wskazuje również etapy po ukończeniu tworzenia struktury aplikacji, tj. optymalizacja kodu oraz testowanie działania aplikacji. Całość prac kończy etap przygotowywania dokumentacji.

Projekt realizowany był z wykorzystaniem systemu kontroli wersji Git. Wszystkie pliki źródłowe projektu znajdują się pod adresem:

<https://github.com/PiotrSmola/Java-Project--Library.git> i będą dostępne do 31.01.2025. Na poniższym rysunku przedstawiono zrzuty ekranu pokazujące historię commitów.





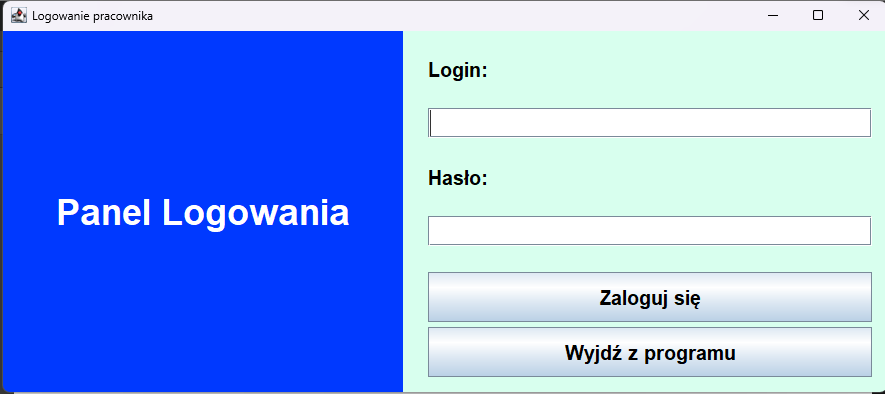
1. **Prezentacja warstwy użytkowej projektu**

Na zrzucie 1 przedstawiono panel logowania użytkownika-pracownika, wyświetlający się natychmiast po uruchomieniu aplikacji.

Na panelu w lewej części okna znajduje się nagłówek informujący użytkownika, że znajduje się w panelu logowania, które po wprowadzeniu poprawnych danych pozwoli zalogować się i uzyskać dostęp do wszystkich funkcji i zasobów aplikacji.

Po prawej stronie okna znajdują się dwa aktywne pola służące do wprowadzania danych logowania użytkownika. Aby w pewnym stopniu zapewnić bezpieczeństwo podczas logowania jako pole ‘Hasło’ zastosowano JPasswordField, aby wpisywane dane przedstawiane były jako kropki.

Pod polami służącymi do wprowadzania danych umieszczono dwa przyciski. Pierwszy ‘Zaloguj się’ służy do przesłania a następnie sprawdzenia poprawności przesłanych danych do metod odpowiedzialnych za walidację danych logowania, które w przypadku powodzenia kończy się przejściem do następnego okna, a w przypadku niepowodzenia wyświetlany jest komunikat „Nieprawidłowy login lub hasło”. Drugi z przycisków ‘Wyjdź z programu’ odpowiedzialny jest za zakończenie procesu aplikacji i jej działania.



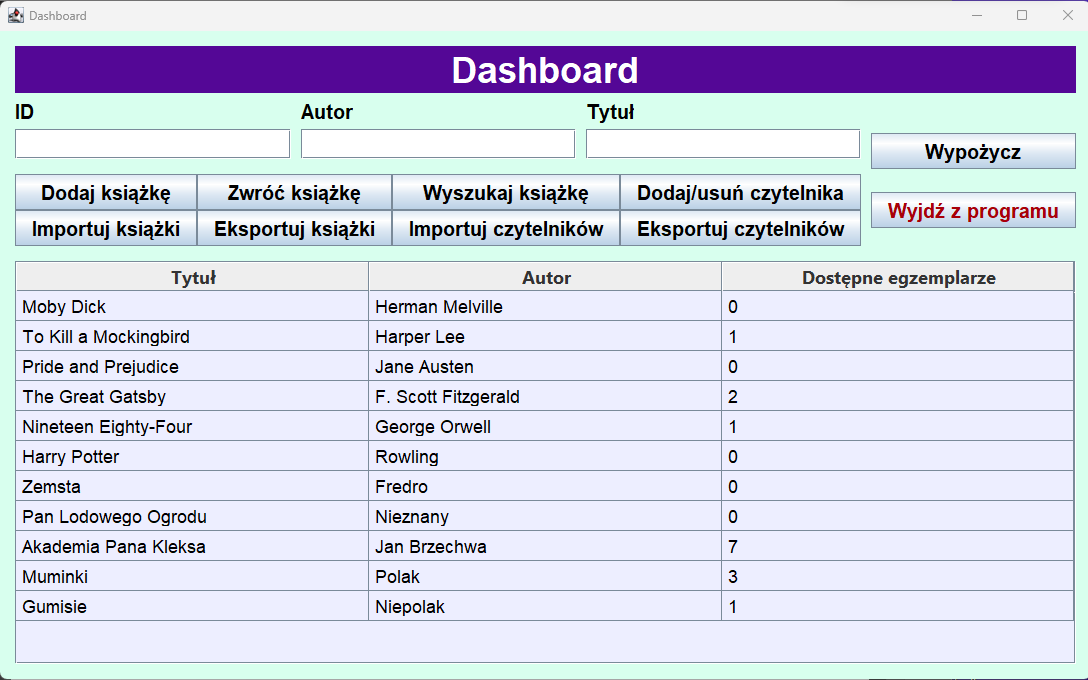
Zrzut 1. Okno logowania pracownika

Na zrzucie 2 przedstawiono główny panel aplikacji ‘Dashboard’, wyświetlający się tylko po uprzednim poprawnym zalogowaniu.

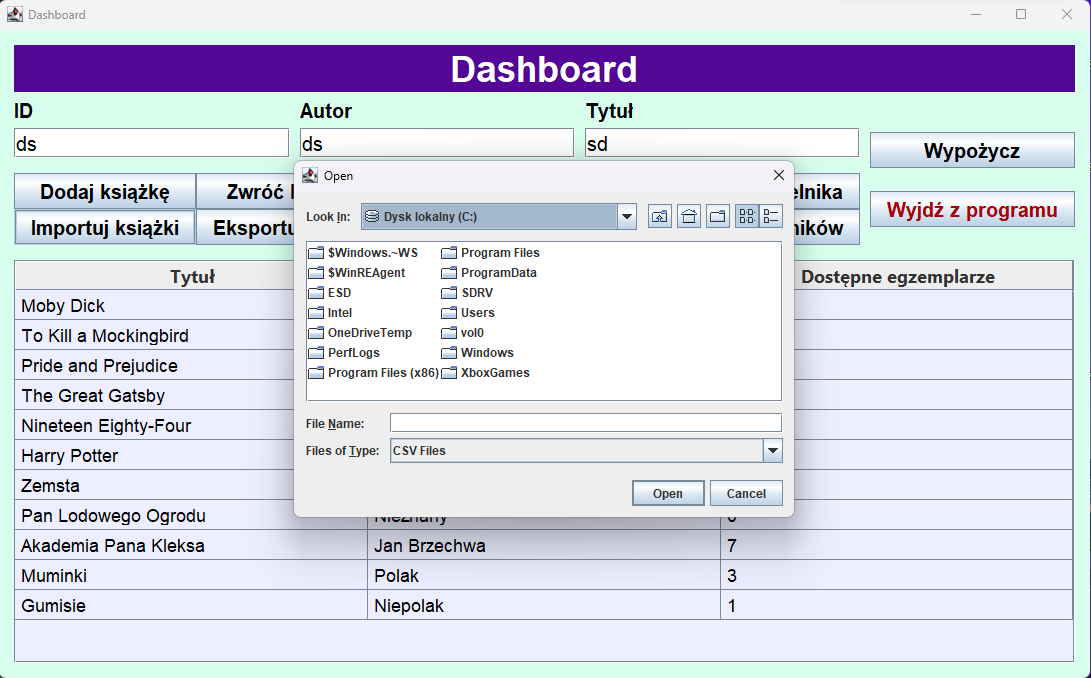
Nagłówek ‘Dashboard’ informuje użytkownika w jakim oknie się obecnie znajduje.

Poniżej widoczne są trzy pola wprowadzania tekstu: ID, Autor, Tytuł oraz przycisk ‘Wypożycz’. Pracownik wprowadza w pola dane służące do wypożyczenia danego tytułu. Aby wypożyczyć książkę, pracownik zobowiązany jest wprowadzić w odpowiednie pola ID czytelnika wygenerowane podczas dodawania go do bazy danych, autora oraz tytuł wypożyczanej książki, aby następnie po kliknięciu przycisku ‘Wypożycz’ przesłać dane do odpowiednich metod, sprawdzających czy możliwe jest żądane wypożyczenie, których wynik prezentowany jest za pomocą odpowiednich komunikatów „Książka została wypożyczona pomyślnie” lub „Nie udało się wypożyczyć książki. Sprawdź wprowadzone dane i spróbuj ponownie”. Aby przyspieszyć proces wprowadzanych danych do wypożyczenia, można w poniższej tabeli, zawierającej wczytane z bazy danych informacje o księgozbiorze, kliknąć na wybrany rekord zawierający dane żądanego tytułu, aby te zostały automatycznie wprowadzone w odpowiednie pola.

Następnie poniżej umieszczony jest zestaw ośmiu przycisków: ‘Dodaj książkę’, ‘Zwróć książkę’, ‘Wyszukaj książkę’, ‘Dodaj/usuń czytelnika’, ‘Importuj książki’, ‘Eksportuj książki’, ‘Importuj czytelników’, ‘Eksportuj czytelników’. Pierwsze cztery powodują uruchomienie kolejnego okna obsługującego odpowiednią z funkcji aplikacji, jednocześnie nie zamykając okna Dashboard. Pozostałe cztery przyciski otwierają okno wyboru ścieżki pliku (Zrzut 3.) z którego chcemy importować dane lub do którego chcemy eksportować dane. Obok wskazanych przycisków znajduje się przycisk ‘Wyjdź z programu’, który kończy działanie aplikacji.



Zrzut 2. Okno Dashboard



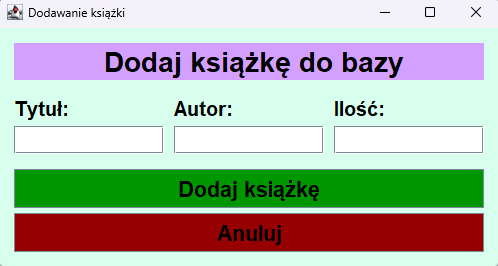
Zrzut 3. Okno wyboru ścieżki pliku eksportu/importu

Na zrzucie 4 przedstawiono okno dodawania książki do bazy danych, do którego dostęp uzyskuje się poprzez wybór odpowiedniej opcji na panelu Dashboard.

W nagłówku ‘Dodaj książkę do bazy’ użytkownik informowany jest w jakim oknie się obecnie znajduje.

Poniżej umieszczone zostały trzy pola do wprowadzania danych: Tytuł, Autor, Ilość, w które pracownik wprowadza dane dotyczące książki dodawanej do bazy danych oraz ilość egzemplarzy danego utworu.

Niżej umieszczono dwa przyciski. Pierwszy ‘Dodaj książkę’ po kliknięciu przekazuje wprowadzone dane do odpowiednich metod sprawdzających czy żądanie jest możliwe do wykonania. W przypadku, gdy wprowadzana książka już istnieje w bazie zwracany jest użytkownikowi komunikat „Książka jest już w bazie”. W przypadku podania poprawnych danych, gdy książka jeszcze nie istnieje w bazie, zwracany jest komunikat „Książka została dodana”, natomiast w przypadku gdy podano błędnie liczbę egzemplarzy zwrócony zostanie komunikat „Nieprawidłowa ilość. Wprowadź liczbę”. Drugi z przycisków - ‘Anuluj’ zamyka okno dodawania książki.



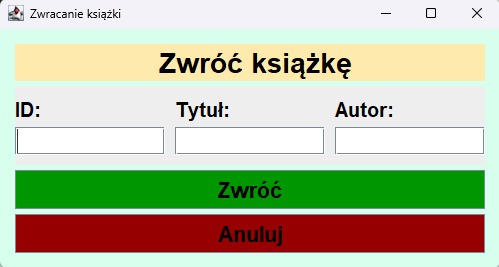
Zrzut 4. Okno dodawania książki do bazy danych

Na zrzucie 5 przedstawiono okno zwracania książki przez czytelnika, do którego dostęp uzyskuje się poprzez wybór odpowiedniej opcji na panelu Dashboard.

W nagłówku ‘Zwróć książkę’ użytkownik informowany jest w jakim oknie się obecnie znajduje.

Poniżej umieszczone zostały trzy pola do wprowadzania danych: ID, Tytuł, Autor, w które pracownik wprowadza ID czytelnika oraz dane dotyczące zwracanej książki.

Niżej umieszczono dwa przyciski. Pierwszy ‘Zwróć’ po kliknięciu przekazuje wprowadzone dane do odpowiednich metod sprawdzających czy żądanie jest możliwe do wykonania. W przypadku, gdy któreś z pól nie zostało uzupełnione zwracany jest komunikat „Wszystkie pola muszą być wypełnione”. W przypadku, gdy nie odnaleziono danego wypożyczenia w bazie zwracany jest komunikat „Nie udało się zwrócić książki. Nieprawidłowy tytuł, ID czytelnika bądź książka nie jest wypożyczona przez tego czytelnika. Spróbuj ponownie”. Natomiast w przypadku, gdy żądanie zakończy się powodzeniem zwracany jest komunikat „Książka zwrócona pomyślnie bez kary.”, jeśli nie wystąpiło opóźnienie w zwrocie książki lub „Książka zwrócona pomyślnie. Kara: …”, gdy takowe opóźnienie wystąpiło. Drugi z przycisków - ‘Anuluj’ zamyka okno zwracania książki przez czytelnika.



Zrzut 5. Okno zwracania książki przez czytelnika

Na zrzucie 6 przedstawiono okno wyszukiwania danego tytułu przez pracownika w bazie danych książek, do którego dostęp uzyskuje się poprzez wybór odpowiedniej opcji na panelu Dashboard.

W nagłówku ‘Wyszukaj książkę’ użytkownik informowany jest w jakim oknie się obecnie znajduje.

Poniżej umieszczone zostały dwa pola do wprowadzania danych: Tytuł, Autor, w które pracownik wprowadza dane dotyczące szukanego tytułu.

pracownik wprowadza ID czytelnika oraz dane dotyczące zwracanej książki.

Niżej umieszczono dwa przyciski. Pierwszy ‘Wyszukaj’ po kliknięciu przekazuje wprowadzone dane do odpowiednich metod sprawdzających czy żądanie jest możliwe do wykonania. W przypadku, gdy któreś z pól nie zostało uzupełnione zwracany jest komunikat „Pola Tytuł i Autor nie mogą być puste”. W przypadku, gdy wprowadzone przez użytkownika dane nie znalazły odzwierciedlenia w bazie książek zwracany jest komunikat „Książka nie została znaleziona”, natomiast jeśli nasze żądanie zakończy się powodzeniem zwracany jest komunikat „Książka została znaleziona. Książka jest dostępna/niedostępna do wypożyczenia”.

Drugi z przycisków - ‘Anuluj’ zamyka okno wyszukiwania książki bazie przez pracownika.



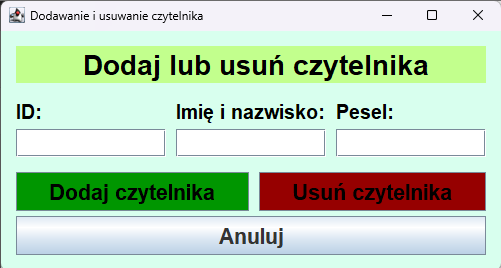
Zrzut 6. Okno wyszukiwania książki w bazie

Na zrzucie 7 przedstawiono okno Dodawania i usuwania danych czytelnika przez pracownika z bazy czytelników, do którego dostęp uzyskuje się poprzez wybór odpowiedniej opcji na panelu Dashboard.

W nagłówku ‘Dodaj lub usuń czytelnika’ użytkownik informowany jest w jakim oknie się obecnie znajduje.

Poniżej umieszczone zostały trzy pola do wprowadzania danych: ID, Imię i nazwisko, Pesel, w które pracownik wprowadza dane dotyczące usuwanego z bazy czytelnika.

Niżej umieszczono trzy przyciski. Pierwszy ‘Dodaj czytelnika’ oraz drugi ‘Usuń czytelnika’ po kliknięciu przekazuje wprowadzone dane do odpowiednich metod sprawdzających czy żądanie jest możliwe do wykonania. W przypadku kliknięcia ‘Dodaj czytelnika’, gdy któreś z pól nie zostało uzupełnione zwracany jest komunikat „Pola Imię i nazwisko oraz Pesel nie mogą być puste” lub jeśli podamy nr Pesel niepoprawnej długości zwrócony zostanie komunikat „Nieprawidłowy numer PESEL”, natomiast w przypadku opcji ‘Usuń czytelnika’ do przeprowadzenia operacji wymagane jest tylko podanie właściwego ID czytelnika, przy czym gdy takowe pole nie zostanie wypełnione zwracany jest komunikat „Pole ID musi być wypełnione” lub jeśli nie odnaleziono w bazie czytelnika o podany ID zwrócony zostanie komunikat „Nie znaleziono czytelnika z ID [wprowadzone ID]”. W przypadku, gdy po wyborze opcji dodawania czytelnika poprawnie uzupełnimy wszystkie wymagane pola zwrócony zostanie komunikat „Czytelnik został dodany pomyślnie. ID: [wygenerowane ID czytelnika]”, natomiast jeśli wybrana opcja usuwania czytelnika zakończy się sukcesem zwracany jest komunikat „Czytelnik został usunięty pomyślnie”. Trzeci z przycisków - ‘Anuluj’ zamyka okno dodawania i usuwania czytelnika.

Zrzut 7. Okno dodawania i usuwania czytelnika

Wszystkie okna aplikacji zostały zaprojektowane w prosty i intuicyjny sposób, aby zapewnić użytkownikom łatwe, szybkie i bezproblemowe działanie, które znacząco wpłynie na optymalizację ich pracy. Każda z funkcji została zaimplementowana tak, aby dowolna z operacji była możliwa do wykonania w jak najszybszy i najprostszy sposób, a każdy błąd lub niepowodzenie zostało oznajmione użytkownikowi w zrozumiały dla niego sposób.

1. **Podsumowanie**

## **Zrealizowane prace**

* Autoryzacja użytkownika

Zaimplementowano system logowania dla pracowników biblioteki, zapewniając bezpieczeństwo i kontrolę dostępu do aplikacji. Pracownik w celu uzyskania dostępu do funkcji programu musi wpisać login i hasło przekazane mu przez administratora bazy pracowników.

* Dashboard

W przypadku pomyślnego zalogowania się użytkownikowi przedstawia się Dashboard, będący głównym oknem systemu. Poza oferowaniem szybkiego wypożyczania oraz wyświetlania wszystkich książek w bazie, pełni również funkcję centrum nawigacyjnego pozwalającego uruchamiać okna obsługujące pozostałe funkcje aplikacji.

* Zarządzanie księgozbiorem

Utworzono system zarządzania księgozbiorem, pozwalający na dodawanie i wyszukiwanie książek. Zintegrowano funkcje wypożyczeń i zwrotów książek, z automatycznym śledzeniem dostępności.

* Obsługa wypożyczeń:

Zaimplementowano logikę rejestrowania informacji o wypożyczeniach, w tym daty wypożyczenia i planowanego zwrotu, umożliwiając monitorowanie i zarządzanie wypożyczonymi książkami.

* Zarządzanie Czytelnikami:

Opracowano system do zarządzania czytelnikami, w tym dodawanie nowych czytelników i usuwanie danych już istniejących.

* Import/eksport danych:

Wprowadzono możliwość importu i eksportu danych księgozbioru i czytelników do/z formatu CSV, co ułatwia migrację danych i ich analizę.

## **Możliwe dalsze prace rozwojowe**

* **Rozbudowa interfejsu użytkownika:** Dalsze prace mogłyby obejmować rozwój bardziej zaawansowanego i intuicyjnego interfejsu użytkownika, z lepszymi możliwościami personalizacji oraz lepszą responsywnością.
* **Integracja z systemami zewnętrznymi:** Możliwe jest rozwijanie integracji z zewnętrznymi systemami bibliotecznymi, takimi jak elektroniczne systemy katalogowania czy systemy zarządzania bibliotekami.
* **Dodanie funkcjonalności raportowania**: Implementacja systemu raportowania, umożliwiająca generowanie szczegółowych analiz i statystyk dotyczących wypożyczeń, zwrotów i preferencji czytelników.
* **Optymalizacja wydajności i skalowalności**: Prace nad optymalizacją wydajności aplikacji, szczególnie w kontekście obsługi dużych zbiorów danych i zwiększonej liczby użytkowników.
* **Zwiększenie bezpieczeństwa danych:** Implementacja dodatkowych warstw zabezpieczeń, szczególnie w kontekście ochrony danych osobowych czytelników i zapobiegania nieautoryzowanemu dostępowi.
* **Zmiana sposobu przechowywania danych:** Zastąpienie bazy danych w plikach profesjonalnym zewnętrznym systemem bazodanowym, na przykład PostgreSQL.

Każdy potencjalnych kierunków dalszego rozwoju mógłby znacząco wpłynąć na funkcjonalność, użyteczność systemu i jego bezpieczeństwo, uczynić go bardziej elastycznym i dostosowanym do potrzeb współczesnej biblioteki.