Specyfikacja wymagań

[BookSwap] ,   
wersja 2024-05-03

|  |  |  |  |
| --- | --- | --- | --- |
|  | Imię nazwisko | Podpis | Data |
| Autor: | Marcin Pilarz | Pilarz | 2024-05-03 |
| Autor: | Kacper Bandyk | Bandyk | 2024-05-03 |

Spis treści

[1. Wstęp 3](#_Toc169261369)

[2. Cele specyfikacji 3](#_Toc169261370)

[3. Zakres specyfikacji 3](#_Toc169261371)

[4. Wymagania funkcjonalne 3](#_Toc169261372)

[5. Wymagania niefunkcjonalne 4](#_Toc169261373)

[5.1 Interfejsy 4](#_Toc169261374)

[5.1.1 Interfejsy użytkownika 4](#_Toc169261375)

[5.1.2 Interfejsy sprzętowe 4](#_Toc169261376)

[5.1.3 Interfejsy komunikacyjne 4](#_Toc169261377)

[5.1.4 Interfejsy programowe 4](#_Toc169261378)

[5.3 Warunki serwisowania (support) 5](#_Toc169261379)

[5.4 Ograniczenia architektury systemu 5](#_Toc169261380)

[5.4 Dokumentacja użytkownika 5](#_Toc169261381)

[5.6 Wymagania licencyjne 6](#_Toc169261382)

[5.7 Prawa autorskie i inne zagadnienia prawne 6](#_Toc169261383)

[6. Wymagania dotyczące jakości modelowanego systemu 6](#_Toc169261384)

[7. Bazy danych 7](#_Toc169261385)

[8. Stosowane standardy 7](#_Toc169261386)

[9. Opis więzów 7](#_Toc169261387)

# 1. Wstęp

Dokument określa wymagania dla aplikacji webowej BookSwap, służącej do wymiany e-booków. Aplikacja umożliwi użytkownikom dodawanie, wymianę oraz przeglądanie   
e-booków w różnych kategoriach. Specyfikacja odwołuje się do standardów webowych   
i protokołów komunikacyjnych używanych w internecie

# 2. Cele specyfikacji

Celem tej specyfikacji jest zdefiniowanie wymagań funkcjonalnych   
i niefunkcjonalnych dla aplikacji BookSwap, platformy internetowej przeznaczonej do wymiany e-booków. Specyfikacja ma na celu zapewnienie jasnych wytycznych dla zespołu projektowego i programistycznego, które pomogą w realizacji aplikacji spełniającej oczekiwania użytkowników końcowych oraz założenia biznesowe.

Aplikacja BookSwap ma za zadanie umożliwić użytkownikom:

* Łatwe i szybkie dodawanie własnych e-booków do systemu w celu ich wymiany.
* Przeglądanie dostępnych e-booków dodanych przez innych użytkowników.
* Wyszukiwanie e-booków według różnych kryteriów, takich jak tytuł, autor lub kategoria.
* Wymianę e-booków na zasadzie bezpośredniego kontaktu z innymi użytkownikami lub poprzez system rekomendacji.
* Zarządzanie własnym kontem i biblioteką e-booków.

# 3. Zakres specyfikacji

Aplikacja umożliwi użytkownikom:

* Rejestrację i logowanie.
* Dodawanie e-booków do systemu.
* Wymianę e-booków z innymi użytkownikami.
* Przeglądanie dostępnych e-booków.
* Otrzymywanie rekomendacji książek.
* Zarządzanie profilami użytkowników.

# 4. Wymagania funkcjonalne

 **Rejestracja i logowanie użytkowników:** Możliwość tworzenia konta oraz logowania   
z użyciem adresu e-mail i hasła.

 **Dodawanie e-booków:** Użytkownicy mogą dodawać e-booki do bazy danych aplikacji, podając tytuł, autora, kategorię, opis oraz plik e-booka.

 **Wyszukiwanie e-booków:** Możliwość wyszukiwania e-booków po tytule, autorze, kategorii.

 **Rekomendacje użytkowników wymieniających:** System rekomendacji użytkowników na podstawie historii wymian między użytkownikami.

 **Profil użytkownika:** Profil zawierający informacje o użytkowniku, jego e-bookach oraz historii wymian.

# 5. Wymagania niefunkcjonalne

 **Interfejs użytkownika:** Intuicyjny i responsywny interfejs, dostosowany do różnych urządzeń.

 **Język:** Aplikacja będzie dostępna w języku polskim.

 **Wydajność i skalowalność:** Aplikacja powinna efektywnie obsługiwać równoczesne żądania od wielu użytkowników.

## 5.1 Interfejsy

Sekcja ta opisuje różne interfejsy, które będą implementowane w aplikacji BookSwap, w celu zapewnienia pełnej funkcjonalności i komunikacji między różnymi komponentami systemu oraz użytkownikami.

### 5.1.1 Interfejsy użytkownika

 **Web UI:** Interfejs użytkownika zostanie zaprojektowany z użyciem React, co pozwoli na dynamiczne aktualizacje bez konieczności przeładowywania strony. Interfejs będzie responsywny, dostosowany do różnych rozmiarów ekranów i urządzeń (desktop, tablet, smartphone).

 **Mobile Friendly:** Projekt responsywny, który zapewni komfortową nawigację i dostępność na urządzeniach mobilnych.

### 5.1.2 Interfejsy sprzętowe

**Kompatybilność urządzeń:** Aplikacja nie będzie wymagać specjalistycznego sprzętu do działania i będzie kompatybilna z ogólnodostępnymi komponentami komputerów osobistych, tabletów i smartfonów.

### 5.1.3 Interfejsy komunikacyjne

 **HTTP/HTTPS:** Komunikacja między front-endem a back-endem będzie realizowana poprzez protokół HTTP, zabezpieczony przez HTTPS.

 **WebSockets:** Dla funkcji wymagających komunikacji w czasie rzeczywistym, takich jak czat między użytkownikami, zostanie użyty protokół WebSockets.

 **REST API:** Back-end będzie eksponował RESTful API, które umożliwi operacje CRUD (Create, Read, Update, Delete) na zasobach takich jak książki, użytkownicy, recenzje itp.

### 5.1.4 Interfejsy programowe

* **REST API:** Front-end aplikacji będzie komunikował się z back-endem poprzez RESTful API, zapewniające standardy komunikacji między różnymi komponentami systemu. API będzie zabezpieczone tokenami JWT (JSON Web Tokens) do zarządzania sesjami i autoryzacją użytkowników.
* **WebSocket:** Do funkcji wymagających komunikacji w czasie rzeczywistym, takich jak czat między użytkownikami, zostaną użyte WebSockety, co umożliwi natychmiastową wymianę wiadomości.

## 5.3 Warunki serwisowania (support)

**Narzędzia do zarządzania i utrzymania systemu:** Panel administracyjny do zarządzania użytkownikami, e-bookami i błędami związanymi z wymianami użytkowników.

## 5.4 Ograniczenia architektury systemu

Specyfikacja techniczna systemu BookSwap zakłada użycie dwóch głównych technologii dla różnych części systemu:

* **Front-end:** Aplikacja kliencka zostanie zbudowana przy użyciu **React**, nowoczesnej biblioteki JavaScript zapewniającej efektywne i interaktywne UI. React został wybrany ze względu na jego dużą społeczność, wsparcie, a także sprawdzone wzorce projektowe, które pomagają w budowie skalowalnych aplikacji webowych.
* **Back-end:** Serwerowa część aplikacji będzie implementowana przy użyciu **Spring Framework**, który jest szeroko stosowany w aplikacjach korporacyjnych. Spring oferuje wszechstronne wsparcie dla budowy bezpiecznych, zarządzalnych   
  i responsywnych usług backendowych w Java. Użyjemy Spring Boot dla uproszczenia konfiguracji i rozwoju mikroserwisów.
* **Baza danych:** System będzie wykorzystywał relacyjne bazy danych (MySQL), które są obsługiwane przez Spring Data JPA, co pozwoli na łatwe zarządzanie relacjami danych i dostęp do danych.
* **Bezpieczeństwo:** Do zabezpieczeń aplikacji wykorzystamy Spring Security, które zapewnia kompleksową ochronę przez uwierzytelnianiem i autoryzacją.
* **Komunikacja międzysystemowa:** Aplikacja będzie używać Spring REST API do obsługi komunikacji między front-endem a back-endem, co umożliwia tworzenie oddzielonych usług, które łatwo się skalują i utrzymują.

## 5.4 Dokumentacja użytkownika

Dokumentacja użytkownika jest niezbędnym elementem systemu BookSwap, zapewniającym wsparcie i przewodnik dla użytkowników na każdym etapie korzystania   
z aplikacji. Dokumentacja ta powinna być jasna, zrozumiała i dostępna bezpośrednio   
z interfejsu aplikacji.

 **Dostępność:** Dokumentacja użytkownika musi być łatwo dostępna z każdej strony aplikacji, poprzez wyraźnie oznaczony link lub przycisk 'Pomoc'.

 **Czytelność:** Teksty pomocy powinny być napisane prostym, zrozumiałym językiem, dostosowanym do poziomu technicznego przeciętnego użytkownika.

 **Język:** Dokumentacja powinna być dostępna przynajmniej w języku polskim, z możliwością rozszerzenia do innych języków w przyszłości.

 **Formatowanie:** Dokumentacja powinna być dobrze zorganizowana, z użyciem nagłówków, list, wyróżnień i obrazów, gdzie to konieczne, aby ułatwić nawigację i zrozumienie.

 **Aktualność:** Dokumentacja musi być regularnie aktualizowana, aby odzwierciedlać zmiany w aplikacji i nowe funkcjonalności.

Specyfikacje 'Pomocy' aplikacji

* **FAQ (Najczęściej Zadawane Pytania):** Sekcja FAQ powinna zawierać odpowiedzi na najczęściej zadawane pytania użytkowników, związane z obsługą aplikacji, problemami technicznymi, procesem wymiany książek i polityką prywatności.
* **Przewodniki użytkownika:** Detaliczne przewodniki krok po kroku opisujące, jak korzystać z głównych funkcji aplikacji, takich jak rejestracja, dodawanie książek, wyszukiwanie i wymiana e-booków.

**Przygotowanie i utrzymanie dokumentacji**

* Dokumentacja użytkownika powinna być przygotowywana i utrzymywana przez zespół specjalistów od UX/UI oraz wsparcia klienta, przy ścisłej współpracy z zespołem technicznym, aby zapewnić jej dokładność i użyteczność. Dostarczenie takiej dokumentacji może znacząco zmniejszyć liczbę zapytań kierowanych do działu obsługi klienta.

## 5.6 Wymagania licencyjne

**Ograniczenia:** Aplikacja musi przestrzegać licencji dotyczących użytych bibliotek oraz  
 e-booków.

## 5.7 Prawa autorskie i inne zagadnienia prawne

**Prawa autorskie:** Aplikacja musi respektować prawa autorskie e-booków.

# 6. Wymagania dotyczące jakości modelowanego systemu

* **Niezawodność (Reliability):** Aplikacja powinna zapewniać wysoką dostępność   
  i minimalizować ryzyko awarii. Testy obciążeniowe komponentów to kluczowe działania w celu zapewnienia niezawodności.
* **Dostępność (Availability):** System powinien być dostępny dla użytkowników przez maksymalną możliwą ilość czasu. Wdrażane będą procedury zapewniające szybkie przywracanie działania usług w przypadku awarii.
* **Użyteczność (Usability):** Interfejs użytkownika powinien być intuicyjny i łatwy   
  w obsłudze dla wszystkich grup użytkowników, z odpowiednimi instrukcjami i pomocą dostępną w ramach aplikacji.
* **Wydajność (Efficiency):** Aplikacja powinna efektywnie zarządzać zasobami systemowymi i zapewniać szybkie czasy odpowiedzi, nawet podczas wysokiego obciążenia.
* **Obsługiwalność (Maintainability):** Kod źródłowy aplikacji powinien być łatwy do zrozumienia, dobrze udokumentowany, co ułatwi przyszłe modyfikacje i aktualizacje.
* **Przenośność (Portability):** System powinien być łatwy do uruchomienia na różnych platformach i systemach operacyjnych bez potrzeby dokonywania znacznych modyfikacji.
* **Bezpieczeństwo danych (Data Security):** Aplikacja musi chronić dane użytkowników przed nieautoryzowanym dostępem i wyciekiem, stosując szyfrowanie, bezpieczne protokoły i regularne audyty bezpieczeństwa.
* **Bezpieczne użycie (Safety):** System powinien chronić użytkowników przed potencjalnie szkodliwymi operacjami i błędami, oferując funkcje takie jak potwierdzenie krytycznych akcji oraz odzyskiwanie danych po awarii.

# 7. Bazy danych

System BookSwap będzie korzystać z relacyjnej bazy danych **MySQL**. Wybrany ze względu na wsparcie dla skomplikowanych zapytań, dobrym zarządzaniem transakcjami oraz wysokim poziomem zabezpieczeń, które są niezbędne do efektywnego zarządzania dużymi ilościami danych i równoczesnym dostępem wielu użytkowników.

Baza danych będzie zawierała następujące główne tabele:

* **Użytkownicy:** Przechowywanie danych o użytkownikach, takich jak identyfikator użytkownika, imię, nazwisko, adres e-mail, hasło (zahashowane).
* **Książki:** Zawiera informacje o każdej książce, w tym identyfikator książki, tytuł, autor, kategoria, opis, ścieżka dostępu do pliku e-booka, rok wydania .
* **Wymiany:** Rejestr transakcji wymian między użytkownikami, w tym identyfikatory książek, identyfikatory użytkowników i daty transakcji.
* **Recenzje:** Opinie użytkowników o wystawiających książki użytkowników   
  i transakcjach wymiany, zawierające oceny i komentarze.
* **Powiadomienia:** Powiadomienia użytkowników, pozwalające wejść w transakcje wymiany z drugim użytkownikiem oraz rozmowy z nim. Atrybuty to : typ wiadomości, wiadomość powiadomienia, data powiadomienia

# 8. Stosowane standardy

W projekcie będą stosowane następujące standardy przemysłowe i technologie:

* **HTML5, CSS3** dla front-endu.
* **ES6+ JavaScript** dla skryptów klienta.
* **Java SE 11+** dla back-endu.
* **JSON** dla formatu wymiany danych.
* **OAuth 2.0** dla autoryzacji.
* **JWT** dla bezpieczeństwa sesji użytkowników.

# 9. Opis więzów

W tej sekcji przedstawione są ograniczenia, które mają wpływ na projekt, rozwój i wdrożenie aplikacji BookSwap. Więzy te wynikają z wyborów technologicznych, wymagań prawnych,   
a także z założeń dotyczących zasobów i infrastruktury.

 **Języki programowania i frameworki:** Aplikacja jest budowana wyłącznie   
z wykorzystaniem React na frontendzie i Spring Framework na backendzie. Ogranicza to wybór dodatkowych technologii i bibliotek, które muszą być kompatybilne z tymi narzędziami.

 **Platformy obsługiwane:** System musi być kompatybilny z najnowszymi wersjami głównych przeglądarek internetowych, co może ograniczyć wykorzystanie niektórych eksperymentalnych funkcji CSS i JavaScript.

 **Prawa autorskie do treści:** Aplikacja musi zapewnić, że wszystkie wymieniane e-booki są dystrybuowane zgodnie z prawem. To wymaga implementacji systemu weryfikacji praw autorskich.

 **Ochrona danych osobowych:** Zgodność z ogólnym rozporządzeniem o ochronie danych (RODO) w UE i innymi lokalnymi przepisami o ochronie danych osobowych. Wymaga to zaimplementowania odpowiednich mechanizmów bezpieczeństwa i prywatności danych.

* **Czas odpowiedzi systemu:** System musi zapewniać odpowiedź na zapytania użytkownika w czasie nie dłuższym niż 2 sekundy, co stawia wymagania wobec wydajności backendu   
  i bazy danych.
* **Dostępność i użyteczność:** System musi być łatwy w obsłudze dla wszystkich grup wiekowych, co wymaga intuicyjnego interfejsu użytkownika oraz dostosowania do osób   
  z różnymi ograniczeniami funkcjonalnymi.