

Kierunek: **Informatyka Techniczna (ITE)**
Specjalność: **Inżynieria Systemów Informatycznych (INS)**

PROJEKT

System do zarządzania biblioteką wykorzystujący głosowy interfejs użytkownika

inż. Bartosz Błyszcz
276951@student.pwr.edu.pl

Prowadzący zajęcia
dr inż. Dariusz Banasiak

1. Wstęp

1.1. Temat i cel projektu

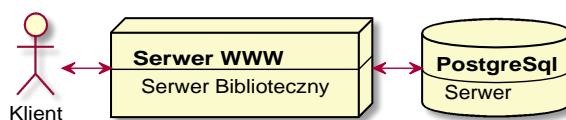
Tematem projektu jest, "System do zarządzania biblioteką wykorzystujący głosowy interfejs użytkownika", w ramach, którego został opracowany wielomodalny interfejs użytkownika służący do obsługi biblioteki. Celem projektu było zbadanie i wykorzystanie możliwości WebSpeechApi [1] w celu umożliwienia poruszania się na stronie internetowej w sposób tradycyjny (klikany) oraz głosowy.

1.2. Założenia techniczne

Tabela 1.1. Technologie wykorzystywane w projekcie
Źródło: Opracowanie własne

Typ aplikacji	Aplikacja internetowe
Język oprogramowania	PHP 7.4 [2]
Baza danych	PostgreSQL [3]
Wykorzystane biblioteki	Symfony 5.4 [4]
	WebSpeechApi [1]
	Bootstrap [5]

W tabeli 1.1 opisano założenia techniczne aplikacji internetowej, jakie przyjęto w trakcie wykonywania aplikacji. Założenia opisują wybrany język oraz technologie, jakie posłużyły w utworzeniu całej aplikacji. Przepływ informacji pokazujących Klientowi został przedstawiony na **diagramie ??**.



Rys. 1.1. Schemat systemu wraz z komunikacją
Źródło: Opracowanie własne

Cały system składa się serwera aplikacji internetowej napisanej w języku PHP 7.4 [2], wykorzystującej framework Symfony 5.4 [4] oraz z serwera bazodanowego wykorzystującego PostgreSQL 10 [3]. Do komunikacji głosowej aplikacja wykorzystuje WebSpeechApi [1], która umożliwia rozpoznawanie języka polskiego i przetwarzanie go na tekst. Dzięki czemu jest możliwość sterowania aplikacją przy pomocy głosu.

2. Dokumentacja techniczna

Poniższy rozdział przedstawia dokumentację techniczną projektu. W projekcie skorzystano z kilku gotowych rozwiązań takich jak:

- **Bootstrap** - jest to framework wykorzystywany do prostszego i szybszego tworzenia responsywnego interfejsu aplikacji internetowej. Dzięki dostarczonym narzędziom pozwala na utrzymywanie raz określonych specyfikacji interfejsu oraz na łatwe ich zmienianie [5],
- **Symfony 5.4** - to bardzo rozbudowany framework dla języka PHP, wykorzystywany do tworzenia skalowalnych i dużych aplikacji internetowych. Framework jest zbudowany z wielu komponentów wielokrotnego użytku, dzięki czemu jest możliwość tworzenia aplikacji na miarę potrzeb bez instalacji niepotrzebnych modułów [4],
- **WebSpeechApi** - jest to biblioteka języka JavaScript, która pozwala na rozpoznawanie oraz syntezę mowy [1].

2.1. Aplikacja internetowa

Serwer aplikacji został napisany przy użyciu języka PHP 7.4 oraz frameworku Symfony 5.4, co pozwoliło na utworzenie aplikacji w architekturze *MVC* (*ang. Model View Controler*). Umożliwia on zarządzanie aplikacją poprzez wykonywanie operacji na bazie danych, z którą jest połączony. Aplikacja służy do zarządzania małą biblioteką, ewentualnie prywatnym zbiorem. Pozwala na:

- dodawanie / usuwanie użytkowników
- dodawanie / usuwanie książek
- zarządzanie wypożyczeniami

Aplikacja jest zabezpieczona panelem logowania, dzięki czemu *użytkownik* o roli administratora może zarządzać zbiorem książek oraz użytkowników. Przykładowy kod został przedstawiony we **fragmentcie 2.1**, pozwala on na uzyskanie informacji o książce oraz wyświetleniu ich użytkownikowi.

```

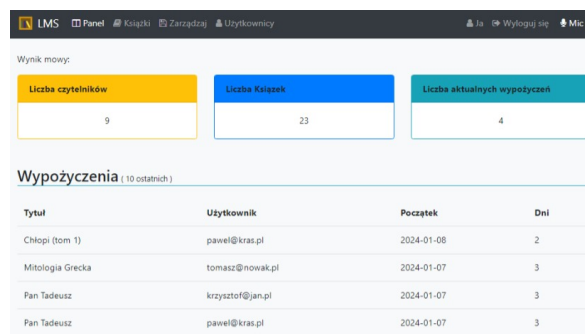
1  /**
2  * @Route("/book/{book}", name="admin_book_site", requirements={"book"="\d{1,9}"})
3  */
4  public function bookSiteAction(Book $book): Response
5  {
6      if ($book->getDeletedAt()) {
7          $this->addFlash('danger', 'Książka została usunięta');
8
9          return $this->redirectToRoute('home');
10     }
11
12     return $this->render('admin/book.html.twig', [
13         'book' => $book,
14     ]);
15 }

```

Fragment kodu 2.1. Metoda bookSiteAction

2.2. Interfejs graficzny

Interfejs graficzny został napisany w językach Html, Css, JavaScript oraz przy użyciu frameworka Bootstrap. Pozwoliło to na stworzenie interfejsu graficznego z działającymi przyciskami, zaciąganych z serwera danymi oraz z prostą kolorystyką co zostało przedstawione na **zdjęciu 2.1**.



The screenshot shows the LMS (Library Management System) interface. At the top, there's a navigation bar with 'LMS', 'Panel', 'Książki', 'Zarządzaj', and 'Użytkownicy'. On the right, there are links for 'Ja', 'Wyloguj się', and a microphone icon. Below the navigation bar, there's a section titled 'Wynik mowy:' with three cards: 'Liczba czytelników' (9), 'Liczba Książek' (23), and 'Liczba aktualnych wypożyczeń' (4). Below this, there's a section titled 'Wypożyczenia (10 ostatnich)' with a table showing the last 10 borrowings.

Tytuł	Użytkownik	Początek	Dni
Chłopi (tom 1)	pawel@kras.pl	2024-01-08	2
Mitologia Grecka	tomasz@nowak.pl	2024-01-07	3
Pan Tadeusz	krzysztof@jan.pl	2024-01-07	3
Pan Tadeusz	pawel@kras.pl	2024-01-07	3

Rys. 2.1. Interfejs graficzny na w aplikacji
Źródło: Opracowanie własne

2.3. Interfejs głosowy

Do obsługi interfejsu wykorzystano WebSpeechApi, który został wykorzystany w dwojaki sposób. Po pierwsze wykorzystano możliwości biblioteki do rozpoznawania i przetwarzania mowy. Dzięki temu po utworzeniu słownika słów kluczowych dostępnego dla aplikacji jest możliwe sterowanie większością funkcjonalności w sposób głosowy. Słownik ten został podany w **tabeli 2.1**. Dodatkowo wykorzystane zostały możliwości syntezy mowy, które najlepiej zostały zobrazowane na formularzu dodawania nowych książek, gdzie po wysłaniu niepoprawnych danych aplikacja zwraca się do użytkownika, opisując mu jakie błędy popełnił.

Tabela 2.1. Słownik słów kluczowych
Źródło: Opracowanie własne

Słownik słów kluczowych			
login	hasło	zaloguj się	ja
wyloguj się	użytkownicy	zarządzaj	książki
panel	dodaj	dodaj nową	dodaj nową książkę
dodaj nowego	tytuł	autor	wiek
granica wieku	zapisz	powrót	książka
usuń	dodaj kopię	edytuj książkę	wypożycz
użytkownik	oddaj	imię	nazwisko
email	data	wyszukaj	indeks
index			

Kod do obsługi tego interfejsu został w pełni napisany w języku Java Script. Kod opisany na fragmencie 2.2 Przedstawia wstępną konfigurację biblioteki WebSpeechApi, która jest wykorzystywana w projekcie.

```

1  speechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition
2  speechGrammarList = window.SpeechGrammarList || window.webkitSpeechGrammarList
3  speechRecognitionEvent = window.SpeechRecognitionEvent || window.webkitSpeechRecognitionEvent
4  grammar = "#JSGF V1.0;";
5  recognition = new speechRecognition();
6  speechRecognitionList = new speechGrammarList();
7  speechRecognitionList.addFromString(grammar, 1);
8  recognition.grammars = speechRecognitionList;
9  recognition.continuous = true;
10 recognition.lang = "pl-PL";
11 recognition.interimResults = false;
12 recognition.maxAlternatives = 1;
13 var recognizing = false;
14 var focus = false;
15 var number = false;
16 var element = null;
17 var book = false;
18 var user = false;
19 var ddate = false;
20 var index = false;
```

Fragment kodu 2.2. Konfiguracja WebSpeechApi

Za rozpoznawanie i przetwarzanie słów na cyfry odpowiada metoda zawarta we fragmencie 2.3.

```

1  textToNumber = (text) => {
2    switch (text) {
3      case "jeden":
4      case " jeden":
5        return 1;
6      case "dwa":
7      case " dwa":
8        return 2;
9      case "trzy":
10     case " trzy":
11       return 3;
12     case "cztery":
13     case " cztery":
```

```

14         return 4;
15     case "ęćpi":
16     case " ęćpi":
17         return 5;
18     case "ścsze":
19     case " ścsze":
20         return 6;
21     case "siedem":
22     case " siedem":
23         return 7;
24     case "osiem":
25     case " osiem":
26         return 8;
27     case "ęćdziewi":
28     case " ęćdziewi":
29         return 9;
30     case "ęćdziesi":
31     case " ęćdziesi":
32         return 10;
33     case "śjedenacie":
34     case " śjedenacie":
35         return 11;
36     case "śdwanacie":
37     case " śdwanacie":
38         return 12;
39     case "śtrzynacie":
40     case " śtrzynacie":
41         return 13;
42     case "śczternacie":
43     case " śczternacie":
44         return 14;
45     case "ęśpitnacie":
46     case " ęśpitnacie":
47         return 15;
48     case "śszesnacie":
49     case " śszesnacie":
50         return 16;
51     case "śsiedemnacie":
52     case " śsiedemnacie":
53         return 17;
54     case "śosiemnacie":
55     case " śosiemnacie":
56         return 18;
57     default:
58         return parseInt(text)
59     }
60 }

```

Fragment kodu 2.3. Metoda zamieniająca słowa na cyfry

Kod ukazany we fragmencie 2.4 odpowiada za konfigurację wydarzeń na stronie, które uruchamiają i wyłączają nasłuchiwanie. Dodatkowo odpowiada za obsłużenie wydarzeń biblioteki.

```

1 recognitionEvent = () => {
2     document.getElementById("mic").onclick = () => {
3         if(!recognizing) {
4             document.getElementById("mic").style.backgroundColor = "red"
5             recognizing = true

```

```

6         recognition.abort();
7         recognition.start();
8     }
9 };
10 recognition.addEventListener("nomatch", () => {
11     document.getElementById("result").innerText = "Nie rozpoznano mowy, rozpocznij na nowo";
12     document.getElementById("mic").style.backgroundColor = "#343a40"
13 });
14 recognition.addEventListener("speechend", () => {
15     document.getElementById("result").innerText = "Wykryto koniec mowy, rozpocznij na nowo";
16     document.getElementById("mic").style.backgroundColor = "#343a40"
17     recognizing = false;
18 });
19 recognition.addEventListener("error", (event) => {
20     document.getElementById("result").innerText = 'Wykryto łąbd: ${event.error}, rozpocznij na nowo';
21     document.getElementById("mic").style.backgroundColor = "#343a40"
22     recognizing = false;
23 });
24 recognition.onstart = function () {
25     recognizing = true;
26 };
27 }

```

Fragment kodu 2.4. Obsługa wydarzeń związanych z mową

Fragment kodu 2.5 odpowiada za wykorzystanie rezultatu przetwarzania mowy na tekst i obsłużenie różnych akcji m.in. uzupełniania formularzy.

```

1 useRecognition((text) => {
2     var tt = text.toLowerCase()
3     document.getElementById("result").innerText = text
4     if (index) {
5         const sss = String(text).trim().slice(0, -1);
6         const _class = 'index-' + textToNumber(sss);
7         document.getElementById(_class).click()
8     }
9     if (focus) {
10         var sss = String(text).trim().slice(0, -1);
11         if (number) {
12             element.value = textToNumber(sss);
13         } else if (ddate) {
14             element.value = Date.parse(String(text).trim());
15         } else {
16             element.value = sss;
17         }
18         element.dispatchEvent(new Event("keyup"))
19         element = null
20         number = false;
21         focus = false
22         return
23     }
24     if (book || user) {
25         var sss = String(text).toLowerCase().trim().slice(0, -1);
26         document.getElementById(sss).click()
27     }
28     tt = tt.slice(0, -1)
29     let utterance = new SpeechSynthesisUtterance("Rozpoznano: " + tt);
30     speechSynthesis.speak(utterance);

```


31 })

Fragment kodu 2.5. Obsługa rezultatu rozpoznawania mowy

Ostatni fragment 2.6 obsługuje słownik słów kluczowych na stronie, wykonując odpowiednie akcje przypisane do poszczególnych słów.

```
1 switch(String(tt).trim()) {
2     case 'login':
3         document.getElementById("inputEmail").focus()
4         focus = true;
5         break;
6     case 'hhaso':
7         document.getElementById("inputPassword").focus()
8         focus = true;
9         break;
10    case 'zaloguj ęsi':
11        document.getElementById("submit").click()
12        break;
13    case 'ja':
14        document.getElementById("Ja").click()
15        break;
16    case 'wyloguj ęsi':
17        document.getElementById("Wyloguj ęsi").click()
18        break;
19    case 'żuytkownicy':
20        document.getElementById("żUytkownicy").click()
21        break;
22    case 'ązarzdzej':
23        document.getElementById("ąZarzdzej").click()
24        break;
25    case 'ążksiki':
26        document.getElementById("ążKsiki").click()
27        break;
28    case 'panel':
29        document.getElementById("Panel").click()
30        break;
31
32    case 'dodaj':
33    case 'dodaj ąnow':
34    case 'dodaj ąnow ążęksik':
35    case 'dodaj nowego':
36        document.getElementById("dodaj").click()
37        break;
38
39    case 'łytu':
40        document.getElementById("book_form_title").focus()
41        element = document.getElementById("book_form_title")
42        focus = true
43        break;
44
45    case 'autor':
46        document.getElementById("book_form_author").focus()
47        element = document.getElementById("book_form_author")
48        focus = true
49        break;
50    case 'wiek':
51        document.getElementById("book_form_ageThreshold").focus()
52        element = document.getElementById("book_form_ageThreshold")
}
```

```

53         focus = true
54         number = true
55         break;
56     case 'granica wieku':
57         document.getElementById("book_form_ageThreshold").focus()
58         element = document.getElementById("book_form_ageThreshold")
59         focus = true
60         number = true
61         break;
62
63     case 'zapisz':
64         document.getElementById("zapisz").click()
65         break;
66     case 'powrót':
67         document.getElementById("powrot").click()
68         break;
69
70     case 'ażksika':
71         document.getElementById("ażksika").style.backgroundColor = 'red'
72         book = true
73         break;
74     case 'ńusu':
75         document.getElementById("book-del").click()
76         break;
77     case 'dodaj ękopi':
78         document.getElementById("book-new-copy").click();
79         break;
80     case 'edytuj ażęksik':
81         document.getElementById("book-edit").click();
82         break;
83     case 'żwypoycz':
84         document.getElementById("new-loan").click();
85         document.getElementById("new-loan").style.backgroundColor = "red";
86         break;
87     case 'żuytkownik':
88         document.getElementById("user").style.backgroundColor = 'red'
89         user = true
90         break;
91     case 'oddaj':
92         document.getElementById("loan-return").click();
93         break;
94     case 'ęimi':
95         document.getElementById("user_form_name").focus();
96         element = document.getElementById("user_form_name");
97         focus = true;
98         break;
99     case 'nazwisko':
100         element = document.getElementById("user_form_surname");
101         document.getElementById("user_form_surname").focus();
102         focus = true;
103         break;
104     case 'email':
105         element = document.getElementById("user_form_email");
106         document.getElementById("user_form_email").focus();
107         focus = true;
108         break;
109     case 'data':
110         element = document.getElementById("user_form_birthDate");

```

```

111         document.getElementById("user_form_birthDate").focus();
112         focus = true;
113         ddate = true;
114         break;
115     case 'wyszukaj':
116         element = document.querySelector("#booksCopies_filter label input");
117         element.focus();
118         focus = true;
119         break;
120     case 'indeks':
121     case 'index':
122         element = document.querySelectorAll(".index");
123         element.forEach(n => {
124             n.style.backgroundColor = "red"
125         })
126         index = true
127         break;
128 }

```

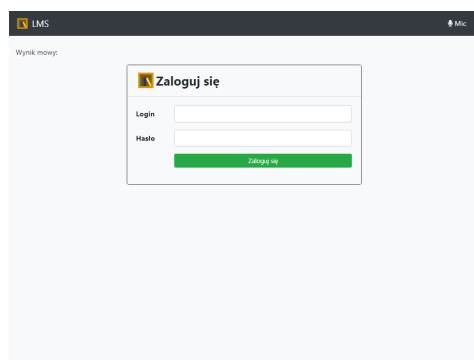
Fragment kodu 2.6. Obsługa słów kluczowych

3. Dokumentacja użytkowa

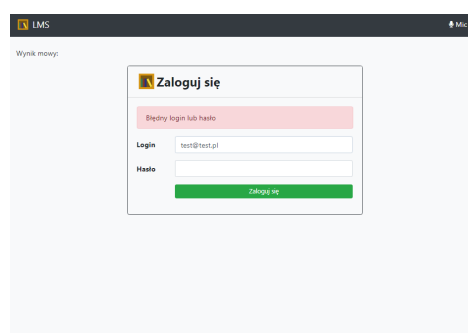
W poniższym rozdziale zostanie zaprezentowana dokumentacja użytkowa aplikacji wytworzonej na potrzeby projektu. W dokumentacji zostaną zaprezentowane poszczególne ekrany oraz interakcje.

3.1. Logowanie

Ekran logowania to pierwszy ekran widoczny dla użytkownika niezalogowanego do systemu. Tak jak wskazano na **rysunku 3.1a** na ekranie występuje prosty formularz pozwalający na wpisanie loginu i hasła użytkownika w celu weryfikacji tożsamości logowanego. Po wciśnięciu przycisku "Zaloguj się" użytkownik zostaje przekierowany do swojego profilu. W przypadku błędnych danych uwierzytelniających użytkownik otrzyma komunikat zwrotny tak jak na **rysunku 3.1b**.

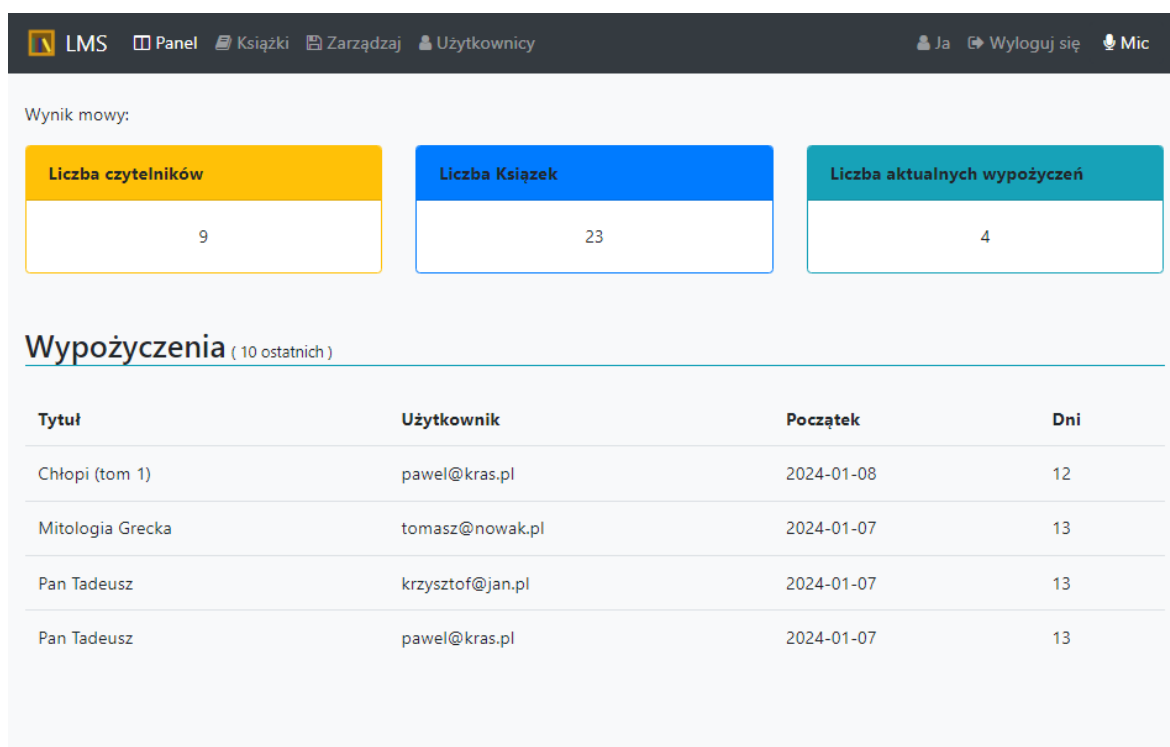


(a) Ekran logowania
Źródło: Opracowanie własne



(b) Ekran logowania - błędne dane
Źródło: Opracowanie własne

Po zalogowaniu, użytkownik przechodzi do ekranu głównego, który jednocześnie jest ekranem ukazującym ogólne statystyki. Został on ukazany na **rysunku 3.2**



Rys. 3.2. Ekran początkowy
Źródło: Opracowanie własne

Na ekranie zostały zaprezentowane takie informacje jak:

- liczba czytelników
- liczba książek
- liczba aktualnych wypożyczeń
- ostatnie 10 wypożyczeń

Pozwala to użytkownikowi zorientować się w aktualnej sytuacji biblioteki.

W górnym prawym rogu ekranu dostępna jest ikona mikrofonu, po której przyciśnięciu aplikacja nasłuchuje wydawanych komend. Po wypowiedzeniu nazwy zakładki z górnego paska użytkownik zostanie do niej przekierowany.

3.2. Książki

Ekran "Książki" pozwala na zarządzanie zbiorem książek. Pierwszy ekran pozwala podejrzeć cały zbiór oraz wyszukiwać konkretne tytuły. Dodatkowo z tego okna można przejść na zakładkę informacyjną książki, oraz dodać nową pozycję. Na tym ekranie po włączeniu opcji *mikrofonu*, można dodać nową książkę bądź wybrać konkretny tytuł. Aby wybrać tytuł należy powiedzieć słowo "Książka", a następnie powiedzieć tytuł książki. Działanie ekranu zostało przedstawione na **rysunkach** 3.3a i 3.3b.

#	Autor	Tytuł	Gatunek	Wiek	Kopie	Dostępność
1	Adam Mickiewicz	Pan Tadeusz	EPIKA	12	3	tak
2	Władysław Stanisław Reymont	Chłopi (tom 1)	EPIKA	12	1	tak
3	Witold Gombrowicz	Ferdydurke	EPIKA	14	1	tak
4	-	Biblia	EPIKA	10	1	tak
5	Jan Parandowski	Mitologia Grecka	EPIKA	8	1	tak
6	Homer	Iliada	EPIKA	10	1	tak
7	Homer	Odyseja	EPIKA	12	1	tak
8	Sofokles	Antygona	EPIKA	14	1	tak
9	Gall Anonim	Kronika polska	EPIKA	10	1	tak
10	William Szekspir	Makbet	DRAMAT	14	1	tak

(a) Ekran Książki
Źródło: Opracowanie własne

#	Autor	Tytuł	Gatunek	Wiek	Kopie	Dostępność
1	Adam Mickiewicz	Pan Tadeusz	EPIKA	12	3	tak
2	Władysław Stanisław Reymont	Chłopi (tom 1)	EPIKA	12	1	tak
3	Witold Gombrowicz	Ferdydurke	EPIKA	14	1	tak
4	-	Biblia	EPIKA	10	1	tak
5	Jan Parandowski	Mitologia Grecka	EPIKA	8	1	tak
6	Homer	Iliada	EPIKA	10	1	tak
7	Homer	Odyseja	EPIKA	12	1	tak

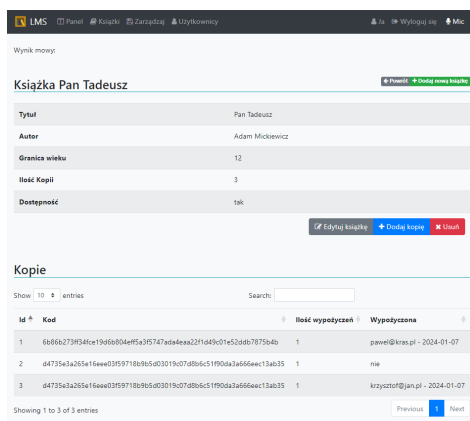
(b) Ekran Książki po wypowiedzeniu słowa książka
Źródło: Opracowanie własne

Po przejściu do ekranu pozwalającego dodać nową książkę, użytkownik zostaje poproszony o wypełnienie krótkiego formularza w którym zawarte są podstawowe informacje o nowej pozycji tak jak to zostało przedstawione na **rysunku 3.4a**. Operacje dodania książki można wykonać również w sposób głosowy, podając etykiety pól formularza a następnie wypowiadając to co ma się w danym polu znajdować.

(a) Formularz dodawania książek
Źródło: Opracowanie własne

(b) Formularz wypełniony błędnie
Źródło: Opracowanie własne

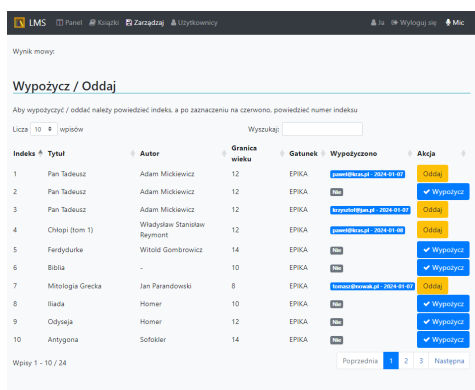
Jeśli po zapisaniu formularza nastąpi błąd. Błędne pola zostaną podkreślone, a błędy zostaną wypisane i powiedziane (z wykorzystaniem syntezy mowy) przez formularz. Wygląd błędnego formularza został ukazany na **ekranie 3.4b**. Po poprawnym wypełnieniu i zapisaniu formularza zostanie dodana nowa książka. Panel informacyjny o książce został wskazany na **obrazku 3.5**. Na tym ekranie dostępne są podstawowe informacje na temat książki oraz informacje o jej wypożyczeniach. Z zaprezentowanego **ekranu 3.5** jest możliwość zarządzać samą książką.



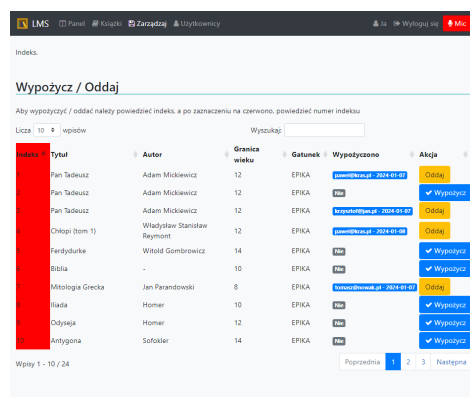
Rys. 3.5. Ekran informacyjny o książce
Źródło: Opracowanie własne

3.3. Zarządzaj

Ekran "Zarządzaj" pozwala na zarządzanie listą wypożyczeń oraz dostępnych książek. Dodatkowo jest to też wyszukiwarka, która pozwala na wyszukiwanie konkretnych tytułów. Został on przedstawiony na **zdjęciu 3.6a**. Ekran pozwala na wypożyczanie oraz oddawanie książek. Można tą operację wykonać również głosowo po wypowiedzeniu słowa "indeks", co zaznaczy kolumnę i poczeka na wypowiedzenie numeru indeksu tak jak na **rysunku 3.6b**. Aby oddać książkę należy wybrać przycisk "Oddaj", bądź wypowiedzieć nazwę indeksu z tym przyciskiem.

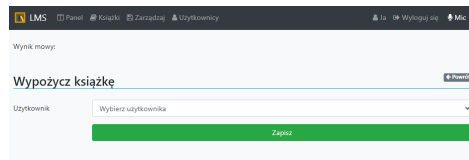


(a) Ekran Zarządzaj
Źródło: Opracowanie własne



(b) Ekran z zaznaczoną kolumną Indeks
Źródło: Opracowanie własne

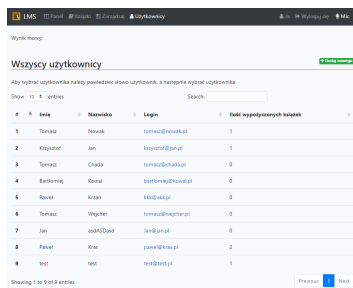
Aby wypożyczyć należy zrobić czynność analogiczną ale dla słowa "Wypożycz". Dodatkowo po wybraniu przycisku "Wypożycz" użytkownik zostanie przekierowany na odpowiedni ekran, który został ukazany na **zdjęciu 3.7**. Tam należy wskazać odpowiedniego użytkownika i kliknąć "Zapisz".



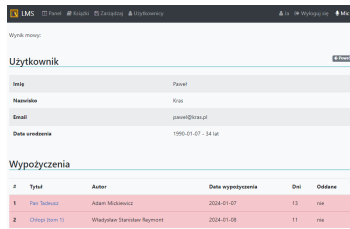
Rys. 3.7. Ekran wypożyczenia
Źródło: Opracowanie własne

3.4. Użytkownicy

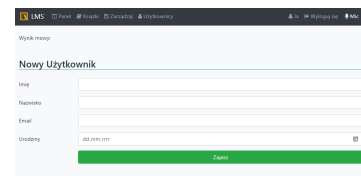
Kolejnym panelem jest panel ”Użytkownicy”, który pozwala na zarządzanie użytkownikami. Główne okno prezentuje listę użytkowników z możliwością dodania nowego użytkownika oraz zobaczenia informacji o użytkowniku. **Rysunek 3.8a** pokazuje listę użytkowników z możliwością wyszukiwania po niej. **Obraz 3.8b** pokazuje okno informacyjne o użytkowniku, a **ekrnn 3.8c** pokazuje formularz dodania nowego użytkownika.



(a) Ekran Użytkownicy
Źródło: Opracowanie własne



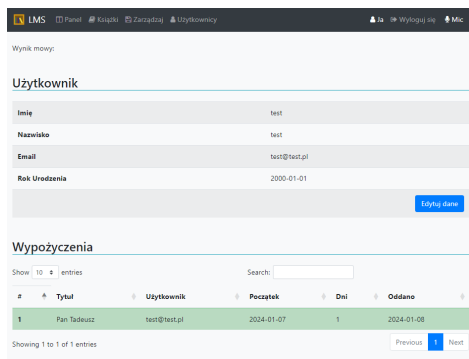
(b) Ekran informacyjny
Źródło: Opracowanie własne



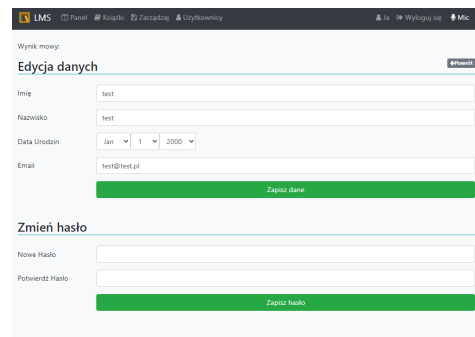
(c) Formularz użytkownika
Źródło: Opracowanie własne

3.5. Ja

Ostatnim panelem jest panel z informacjami zalogowanego użytkownika, pokazany na **rysunku 3.9a**. Pozwala on na sprawdzenie informacji o koncie oraz ich zmodyfikowaniu za pomocą formularza pokazanego na **zdjęciu 3.9b**.



(a) Ekran Ja
Źródło: Opracowanie własne



(b) Formularz modyfikacji danych
Źródło: Opracowanie własne

4. Wnioski

W trakcie tego projektu zostały wykorzystane biblioteka WebSpeechApi do języka JavaScript. Jest to bardzo prosta w użyciu biblioteka dająca duże możliwości i prosta w obsłudze. Dużym plusem tej biblioteki jest możliwość skorzystania z możliwości przeglądarek do obsługi rozpoznawania i syntezy mowy. Pozwoliło to na unowocześnienie aplikacji prostej biblioteki przez dodanie do niej obsługi interfejsu głosowego. Dzięki czemu osoby niepełnosprawne z problemami ruchowymi mogą obsługiwać aplikację poprzez wykorzystanie głosu. Kolejnym atutem jest możliwość syntezy głosu, dzięki czemu jest możliwość zwracania informacji zwrotnej tak jak zostało to pokazane na formularzu dodawania książek. Biblioteka bardzo dobrze sobie radzi z tekstem ciągłym i znakami specjalnymi np. przy dyktowaniu adresu email. Jednakże zdania należało wypowiadać bardzo wyraźnie ponieważ samo rozpoznawanie mowy nie zawsze "rozumiało" celu wypowiedzi albo słów, wypowiedzianych jeśli były robione w sposób niezrozumiały. Możliwość skorzystania z tych możliwości biblioteki wskazuje jak duży potencjał mają aplikacje przeglądarkowe, które mogą być uruchamiane bezpośrednio w przeglądarce bądź jako PWA (ang. *Progressive Web App*) czyli aplikacja "instalowana" lokalnie która może w pewnym stopniu działać bez dostępu do internetu.

Wykaz rysunków

1.1	Schemat systemu wraz z komunikacją	3
2.1	Interfejs graficzny na w aplikacji	5
3.2	Ekran początkowy	13
3.5	Ekran informacyjny o książce	15
3.7	Ekran wypożyczenia	16

Bibliografia

- [1] WebSpeechApi. „*WebSpeechApi*”. URL: <https://dvcs.w3.org/hg/speech-api/raw-file/tip/webspeechapi> (term. wiz. 2024-01-10).
- [2] PHP. „*PHP*”. URL: <https://www.php.net/> (term. wiz. 2023-01-10).
- [3] PostgreSQL. „*Postgresql*”. URL: <https://www.postgresql.org/pl/> (term. wiz. 2023-01-10).
- [4] Symfony. „*Symfony*”. URL: <https://www.symfony.com/> (term. wiz. 2023-01-10).
- [5] Bootstrap. „*Bootstrap*”. URL: <https://www.getbootstrap.com/> (term. wiz. 2023-01-10).