

Politechnika Warszawska
Wydział Elektryczny
Kierunek Informatyka Stosowana

OPIS SYSTEMU DO ZARZĄDZANIA PUBLIKACJAMI NAUKOWYMI

Piotr Jeleniewicz
nr indeksu 291072

Spis treści

1	Wstęp	2
2	Opis działania	2
3	Analiza funkcjonalna	2
4	Architektura techniczna	3
4.1	Diagram komponentów	3
4.2	Opis aplikacji serwerowej	4
4.2.1	Metody analizy plików PDF	5
4.3	Opis aplikacji klienckiej(mobilnej)	5

1 Wstęp

W ramach projektu dyplomowego wykonany zostanie system do zarządzania publikacjami naukowymi, który będzie automatycznie dokonywał analizy plików PDF. Będzie się on składał z aplikacji serwerowej służącej do analizy plików oraz obsługi bazy danych, która napisana będzie przy użyciu Node.js oraz języka TypeScript, a także aplikacji klienckiej pracującej na urządzeniach z systemem Android napisanej w języku Kotlin.

2 Opis działania

Po zainstalowaniu aplikacji na urządzeniu z systemem Android oraz jej uruchomieniu, użytkownik zostanie poproszony o zalogowanie się lub utworzenie konta. Po zalogowaniu, użytkownika będzie miał możliwość tworzenia nowych publikacji, edytowania istniejących oraz wyszukiwania innych użytkowników oraz ich publikacji.

3 Analiza funkcjonalna

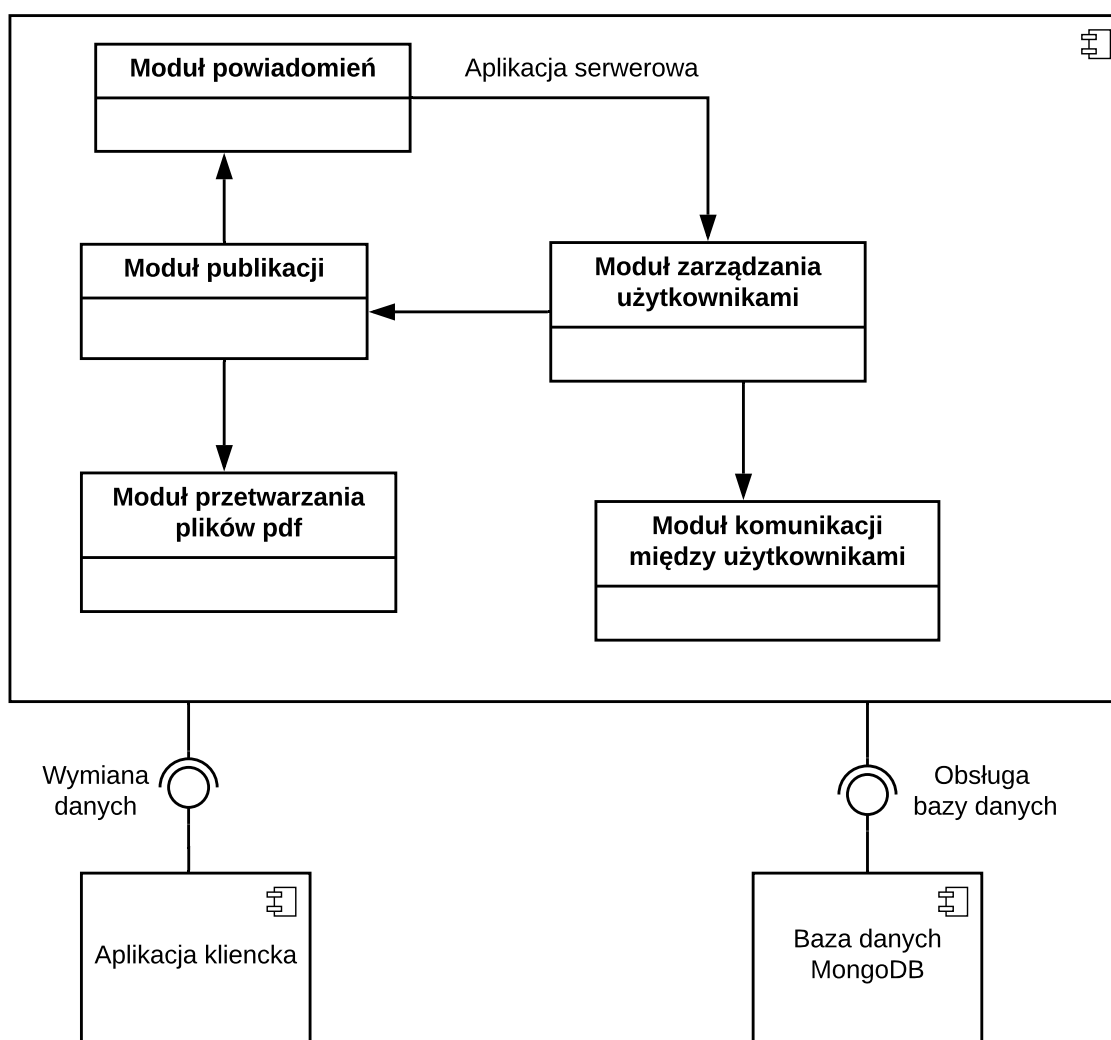
System będzie udostępniał następujące funkcjonalności:

1. Wyświetlanie publikacji naukowych.
2. Wyszukiwanie publikacji naukowych na podstawie autorów oraz nazwy.
3. Tworzenie nowej publikacji w oparciu o metadane dostępne w pliku PDF.
4. Obserwowanie innych użytkowników.
5. Edycja publikacji.

6. Wysyłanie powiadomień o nowo dodanej publikacji do osób obserwujących autora publikacji.
7. Możliwość komentowania publikacji.
8. Komunikacja tekstowa z innymi użytkownikami systemu.

4 Architektura techniczna

4.1 Diagram komponentów



4.2 Opis aplikacji serwerowej

Aplikacja serwerowa zostanie wykonana przy następujących założeniach:

1. Wykorzystany zostanie framework Node.js oraz język TypeScript.
2. Aplikacja będzie udostępniać REST Api do komunikacji z aplikacją kliencką.
3. Wraz bazą danych będzie uruchamiana w środowisku Docker.
4. Do przechowywania danych wykorzystana zostanie baza danych MongoDB.
5. Będzie się składać z następujących modułów:
 - (a) moduł zarządzania użytkownikami - odpowiedzialny za rejestrację i logowanie użytkowników,
 - (b) moduł komunikacji między użytkownikami - odpowiedzialny za wysyłanie wiadomości pomiędzy użytkownikami aplikacji,
 - (c) moduł publikacji - odpowiedzialny za dodawanie, edycję oraz usuwanie aplikacji,
 - (d) moduł przetwarzania plików pdf - odpowiedzialny za analizę metadanych plików pdf i tworzeniu na ich podstawie opisu publikacji,
 - (e) moduł powiadomień - odpowiedzialny za wysyłanie powiadomień do użytkowników,
6. System będzie zachowywał standardy bezpieczeństwa takie jak wykorzystanie protokołu HTTPS zamiast HTTP oraz walidacja danych otrzymanych w żądaniach.

4.2.1 Metody analizy plików PDF

Jedną z najistotniejszych funkcji tego systemu będzie analiza plików PDF w celu pobrania z nich metadanych określających informacje takie jak tytuł czy autor. Przetestowane zostały następujące rozwiązania:

1. Moduł *pdf-parser* - moduł najpopularniejszy ze wszystkich analizowanych, ale też nie rozwijany od 2 lat;
2. Moduł *pdfreader* - moduł najmniej popularny ze wszystkich analizowanych, wciąż rozwijany;
3. Moduł *pdf2json* - moduł dość popularny, wciąż rozwijany;

Podjęta została próba z wykorzystaniem każdego z modułów. Niestety moduły *pdfreader* oraz *pdf2json* pozbawione są typów, wykrozystywanych w języku TypeScript, w którym tworzona będzie aplikacja serwerowa. Dlatego też do analizy plików pdf wykorzystany zostanie moduł *pdf-parser*, który posiada pełne wsparcie typów w języku TypeScript. Za pomocą tego modułu, uzyskujemy informacje o pliku pdf takie jak tytuł, autor, i inne związane z wersją pdf czy słowami kluczowymi. Umożliwia on także odczyt tekstu z pliku pdf, co może być przydatne do tworzenia skrótów zawierających wstęp pliku.

4.3 Opis aplikacji klienckiej(mobilnej)

Aplikacja kliencka zostanie wykonana jako aplikacja mobilna przy następujących założeniach:

1. Aplikacja będzie pracować na systemie Android.
2. Wykorzystany zostanie język Kotlin.
3. Wraz bazą danych będzie uruchamiana w środowisku Docker.
4. Do obsługi powiadomień wykorzystany zostanie Firebase Cloud Messaging.