

Politechnika Warszawska

W Y D Z I A Ł   E L E K T R Y C Z N Y



Instytut Elektrotechniki Teoretycznej  
i Systemów Informacyjno-Pomiarowych  
Zakład Elektrotechniki Teoretycznej  
i Informatyki Stosowanej

# Praca dyplomowa inżynierska

na kierunku Informatyka stosowana  
w specjalności Inżynieria oprogramowania

Aplikacja do zarządzania publikacjami naukowymi z  
automatyczną analizą PDF

**Piotr Jeleniewicz**

nr albumu 291072

promotor  
dr inż. Bartosz Chaber

WARSZAWA 2021

# APLIKACJA DO ZARZĄDZANIA PUBLIKACJAMI NAUKOWYMI Z AUTOMATYCZNĄ ANALIZĄ PDF

## Streszczenie

Praca składa się z

**Słowa kluczowe:** praca dyplomowa, LaTeX, jakość

# REFERENCE MANAGER WITH AUTOMATIC PDF ANALYSIS

## Abstract

This thesis presents.

**Keywords:** thesis, LaTeX, quality

WARSZAWA, 1 lutego 2021

POLITECHNIKA WARSZAWSKA  
WYDZIAŁ ELEKTRYCZNY

### OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa inżynierska pt. Aplikacja do zarządzania publikacjami naukowymi z automatyczną analizą PDF:

- została napisana przeze mnie samodzielnie,
- nie narusza niczyich praw autorskich,
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej. Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Piotr Jeleniewicz.....



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Założenia projektowe . . . . .	1
<b>2</b>	<b>Przedstawienie wykorzystywanych technologii</b>	<b>3</b>
2.1	Aplikacja serwerowa . . . . .	3
2.1.1	Node.js . . . . .	4
2.1.2	TypeScript . . . . .	4
2.1.3	MongoDB . . . . .	5
2.1.4	Docker . . . . .	5
2.2	Aplikacja kliencka . . . . .	6
2.2.1	Android . . . . .	6
2.2.2	Kotlin . . . . .	6
	<b>Bibliografia</b>	<b>7</b>

# Podziękowania

Piotr Jeleniewicz

# Rozdział 1

## Wstęp

Przy pisaniu artykułów naukowych często sięga się po efekty prac przedstawione w innych publikacjach naukowych. Wraz z pisanem coraz większej liczby tego typu dokumentów, ilość wykorzystywanych w nich pozycji bibliograficznych może zacząć znacznie wzrastać co może utrudnić odnajdowanie potrzebnych publikacji w stale rozszerzającym się ich zbiorze.

Dlatego też ta praca będzie przedstawiała efekty prac nad aplikacją do zarządzania publikacjami naukowymi z automatyczną analizą plików PDF, której głównym celem jest ułatwienie procesu zarządzania publikacjami naukowymi, które przechowywane są w formie plików PDF. System będzie składał się z aplikacji klienckiej przygotowanej dla systemu Android oraz aplikacji serwerowej działającej w kontenerze Dockera.

### 1.1 Założenia projektowe

System powstający w ramach pracy inżynierskiej będzie opierał się o następujące założenia:

1. Aplikacja serwerowa będzie napisana przy użyciu Node.js oraz języka TypeScript.
2. W celu ułatwienia konfiguracji środowiska deweloperskiego jak i produkcyjnego, baza danych wraz z aplikacją serwerową będą uruchamiane w sposób skonteneryzowany przy wykorzystaniu technologii Docker.
3. Aplikacja kliencka będzie przeznaczona na system Android oraz do jej napisania wykorzystany zostanie język Kotlin.
4. W celu korzystania z aplikacji użytkownik będzie musiał utworzyć konto w systemie.

5. Do uzyskania informacji z pliku PDF dotyczących publikacji wykorzystane zostanie API dostępne pod adresem:  
<https://api.crossref.org/>
6. Aplikacje będą tworzone w sposób modułowy, umożliwiając stosunkowo łatwą możliwość rozbudowywania funkcjonalności.
7. Podczas implementacji obu aplikacji w miarę możliwości stosowane będą najlepsze praktyki programistyczne.

W systemie będą dostępne następujące funkcjonalności:

1. Wyświetlanie listy publikacji.
2. Wyświetlanie opisu publikacji naukowych.
3. Tworzenie nowej publikacji w oparciu o metadane oraz numer DOI, jeśli są dostępne w dodawanym pliku PDF.
4. Edycja publikacji.
5. Pobranie pliku PDF powiązanego z daną publikacją.



## Rozdział 2

# Przedstawienie wykorzystywanych technologii

Przed rozpoczęciem implementacji projektu przeprowadzona została szczegółowa analiza dostępnych technologii zarówno w kontekście aplikacji serwerowej jak również klienckiej. W trakcie jej trwania pod uwagę brane były przede wszystkim aspekty dotyczące specyfik danych technologii takich jak wydajność czy też sugerowane przeznaczenie poszczególnych rozwiązań, ale także kwestie dotyczące osobistych preferencji odnoszących się do danych technologii.

### 2.1 Aplikacja serwerowa

W obecnych czasach ilość technologii pozwalających na pisanie aplikacji serwerowych jest ogromna. Podjęcie jednoznacznego wyboru dotyczącego, której z nich należy użyć do danego zadania jest niemalże niemożliwe, jednakże każda z nich cechuje się swoimi indywidualnymi cechami, które pozwalają w pewnym stopniu ocenić, które z dostępnych narzędzi będzie odpowiednie do rozwiązania wybranego problemu. W przypadku aplikacji serwerowej, tworzonej w ramach tej pracy, głównymi cechami brnymi pod uwagę była wydajność danej technologii oraz ilość kodu wymagana do poprawnego i bezpiecznego działania aplikacji. Po przeanalizowaniu dostępnych rozwiązań, aplikacja zostanie napisana w środowisku Node.js, przy użyciu języka TypeScript. Do przechowywania danych wykorzystana zostanie baza danych MongoDB. Cała aplikacja serwerowa będzie uruchomiona jako kontenery Dockera.

### 2.1.1 Node.js

Node.js jest środowiskiem uruchomieniowym bazującym na koncepcie „*JavaScript everywhere*”, który unifikuje język używany podczas procesu tworzenia aplikacji webowej, wykorzystując JavaScript zarówno po stronie klienta jak i serwera. W związku z tym, że w ramach projektu aplikacja kliencka będzie stworzona w formie aplikacji mobilnej na system Android, koncept ten nie zostanie spełniony. jednakże przy tworzeniu hipotetycznej aplikacji klienckiej działającej w przeglądarkach internetowych wykorzystywanie jednego języka, jest sytuacją ułatwiającą rozwój i utrzymania takiej aplikacji.

Do niewątpliwych zalet środowiska Node.js należy ogromna ilość bibliotek dostępnych do wykorzystania w aplikacji. Dzięki temu wiele problemów może zostać rozwiązanych przy użyciu gotowych komponentów, co znacznie wpływa na zmniejszenie ilości kodu, który jest wymagany do rozwiązania konkretnego problemu. Z punktu widzenia aplikacji serwerowej tworzonej w ramach tego projektu, niezwykle istotna okazała się dostępność biblioteki umożliwiającej przetwarzanie i odczyt danych z plików *pdf*.

Inną ważną cechą środowiska Node.js, w odróżnieniu od środowisk wykorzystujących język Python, jest bardzo dobra obsługa standardu JSON, która pozwala na używanie obiektów zgodnych ze standardem JSON niemal jako standardowych obiektów języka JavaScript, co znacznie wpływa na ułatwienie komunikacji pomiędzy aplikacją serwerową a kliencką.

### 2.1.2 TypeScript

Językiem który zostanie wykorzystany do napisania aplikacji serwerowej będzie TypeScript. Jest on nadzbiorem języka JavaScript, utworzonym i utrzymanym przez firmę Microsoft. Zasadniczą różnicą pomiędzy tymi językami jest typowanie.

Typowanie można podzielić między innymi na następujące dwa rodzaje:

- Typowanie statyczne - zmienne mają ustalany typ w momencie deklaracji. Wykorzystywane w języku TypeScript
- Typowanie dynamiczne - typ zmiennej wynika z wartości jaka jest w niej przechowywana. Wykorzystywane w języku JavaScript.

Pomimo że typowanie dynamiczne pozwala na większą elastyczność przy wykorzystaniu zmiennych, typowanie statyczne pozwala na wykrycie części błędów w momencie kompilacji, bądź też transpilacji kodu.

W celu uruchomienia kodu napisanego w języku TypeScript, musi najpierw zostać dokonana jego transpilacja do języka JavaScript. Dopiero ten kod jest realnie wykonywany w środowisku uruchomieniowym.

### 2.1.3 MongoDB

MongoDB jest nierelacyjną bazą danych, wykorzystywanym w ramach aplikacji serwerowej. W systemach nierelacyjnych w przeciwieństwie od relacyjnych, informacje nie są przechowywane w tabelach, a w formie dokumentów w stylu JSON, które oprócz samych danych przechowują także ich schemat. Dzięki temu zmiana schematu danych, nie wymaga zazwyczaj skomplikowanych migracji i pozwala na bardzo dynamiczne zarządzanie strukturą danych. Wadą takiego rozwiązania jest utrudnione wyszukiwanie żądanych informacji, jeśli wymagają one bardziej skomplikowanych kwerend, a także znacznie mniejsza szybkość wyszukiwania, zwłaszcza w przypadku większej liczby danych.

W celu łączenia się z bazą danych i mapowania obiektów z bazy danych na obiekty języka JavaScript wykorzystywany jest ODM Mongoose. Dzięki temu że baza danych przechowuje dokumenty w stylu JSON, w bardzo prosty sposób można zapisywać obiekty otrzymane od klienta w ramach żądania HTTP do bazy danych, a także wysyłać klientowi obiekty z bazy, bez konieczności znaczącej ingerencji w ich format i strukturę.

### 2.1.4 Docker

W celu ułatwienia procesu wdrożenia aplikacji serwerowej w środowisku produkcyjnym, aplikacja wraz z bazą danych zostały zamknięte w kontenery Dockera. Jest to oprogramowanie umożliwiające uruchamianie skonteneryzowanych aplikacji w środowisku odizolowanym od głównego systemu operacyjnego. Kontener zawiera w sobie całe środowisko potrzebne do uruchomienia aplikacji, w odróżnieniu jednak od typowej maszyny wirtualnej kontener nie musi zazwyczaj zawierać w sobie pełnego systemu operacyjnego co znacznie poprawia wydajność takiego rozwiązania, a jednocześnie sprawia że jesteśmy w stanie uruchomić aplikację zamkniętą w kontenerze, bez konieczności wstępnej konfiguracji środowiska uruchomieniowego, co znacznie poprawia przenośność takiej aplikacji. Izolacja kontenerów pomiędzy sobą, a także od pozostałych procesów głównego systemu operacyjnego może dodatkowo mieć wpływ na podniesienie bezpieczeństwa. Komunikacja pomiędzy różnymi kontenerami może mieć miejsce tylko w specjalnie przygotowanych kanałach komunikacyjnych, np. poprzez wydzieloną sieć, w skład której wchodzi tylko te kontenery, które muszą pomiędzy sobą wymieniać dane. Taka sytuacja ma miejsce przy wymianie danych pomiędzy aplikacją serwerową przygotowywaną w ramach tej pracy, a bazą danych MongoDB.

## **2.2    Aplikacja kliencka**

### **2.2.1    Android**

### **2.2.2    Kotlin**

# Bibliografia

- [1] daemon9, „LOKI2”, Phrack Magazine, Issue 51. <http://phrack.org>
- [2] van Hauser, Reverse WWW Shell, THC, The Hacker’s Choice.  
[www.thc.org](http://www.thc.org)



## **Opinia**

o pracy dyplomowej magisterskiej wykonanej przez dyplomanta

**Zdolnego Studenta i Pracowitego Kolegę**

Wydział Elektryczny, kierunek Informatyka, Politechnika Warszawska

Temat pracy

**TYTUŁ PRACY DYPLOMOWEJ**

Promotor: **dr inż. Miły Opiekun**

Ocena pracy dyplomowej: **bardzo dobry**

### **Treść opinii**

Celem pracy dyplomowej panów dolnego Studenta i Pracowitego Kolegi było opracowanie systemu pozwalającego symulować i opartego o oprogramowanie o otwartych źródłach (ang. Open Source). Jak piszą Dyplomanci, starali się opracować system, który łatwo będzie dostosować do zmieniających się dynamicznie wymagań, będzie miał niewielkie wymagania sprzętowe i umożliwiał dalszą łatwą rozbudowę oraz dostosowanie go do potrzeb. Przedstawiona do recenzji praca składa się z krótkiego wstępu jasno i wyczerpująco opisującego oraz uzasadniającego cel pracy, trzech rozdziałów (2-4) zawierających opis istniejących podobnych rozwiązań, komponentów rozpatrywanych jako kandydaci do tworzonego systemu i wreszcie zagadnień wydajności wirtualnych rozwiązań. Piąty rozdział to opis przygotowanego przez Dyplomantów środowiska obejmujący opis konfiguracji środowiska oraz przykładowe ćwiczenia laboratoryjne. Ostatni rozdział pracy to opis możliwości dalszego rozwoju projektu. W ramach przygotowania pracy Dyplomanci zebrali i przedstawili w bardzo przejrzysty sposób duży zasób informacji, co świadczy o dobrej orientacji w nowoczesnej i ciągle intensywnie rozwijanej tematyce stanowiącej zakres pracy i o umiejętności przejrzystego przedstawienia tych wyników. Praca zawiera dwa dodatki, z których pierwszy obejmuje wyniki eksperymentów i badań nad wydajnością, a drugi to źródła skryptów budujących środowisko.

Dyplomanci dość dobrze zrealizowali postawione przed nimi zadanie, wykazali się więc umiejętnością zastosowania w praktyce wiedzy przedstawionej w rozdziałach 2-4. Uważam, że cele postawione w założeniach pracy zostały pomyślnie zrealizowane. Proponuję ocenę bardzo dobrą (5).

(data, podpis)

## **Recenzja**

pracy dyplomowej magisterskiej wykonanej przez dyplomanta

**Zdolnego Studenta i Pracowitego Kolegę**

Wydział Elektryczny, kierunek Informatyka, Politechnika Warszawska

Temat pracy

**TYTUŁ PRACY DYPLOMOWEJ**

Recenzent: **prof. nzw. dr hab. inż. Jan Surowy**

Ocena pracy dyplomowej: **bardzo dobry**

### **Treść recenzji**

Celem pracy dyplomowej panów dolnego Studenta i Pracowitego Kolegi było opracowanie systemu pozwalającego symulować i opartego o oprogramowanie o otwartych źródłach (ang. Open Source). Jak piszą Dyplomanci, starali się opracować system, który łatwo będzie dostosować do zmieniających się dynamicznie wymagań, będzie miał niewielkie wymagania sprzętowe i umożliwiał dalszą łatwą rozbudowę oraz dostosowanie go do potrzeb. Przedstawiona do recenzji praca składa się z krótkiego wstępu jasno i wyczerpująco opisującego oraz uzasadniającego cel pracy, trzech rozdziałów (2-4) zawierających bardzo solidny i przejrzysty opis: istniejących podobnych rozwiązań (rozdz. 2), komponentów rozpatrywanych jako kandydaci do tworzonego systemu (rozdz. 3) i wreszcie zagadnień wydajności wirtualnych rozwiązań, zwłaszcza w kontekście współpracy kilku elementów sieci (rozdział 4). Piąty rozdział to opis przygotowanego przez Dyplomantów środowiska obejmujący opis konfiguracji środowiska oraz przykładowe ćwiczenia laboratoryjne (5 ćwiczeń). Ostatni, szósty rozdział pracy to krótkie zakończenie, które wylicza także możliwości dalszego rozwoju projektu. W ramach przygotowania pracy Dyplomanci zebrali i przedstawili w bardzo przejrzysty sposób duży zasób informacji o narzędziach, Rozdziały 2, 3 i 4 świadczą o dobrej orientacji w nowoczesnej i ciągle intensywnie rozwijanej tematyce stanowiącej zakres pracy i o umiejętności syntetycznego, przejrzystego przedstawienia tych wyników. Drobne mankamenty tej części pracy to zbyt skrótowe omawianie niektórych zagadnień technicznych, zakładające dużą początkową wiedzę czytelnika i dość niestaranne podejście do powołań na źródła. Utrudnia to w pewnym stopniu czytanie pracy i zmniejsza jej wartość dydaktyczną (a ta zdaje się być jednym z celów Autorów), ale jest zrekompensowane zawartością merytoryczną. Praca zawiera dwa dodatki, z których pierwszy obejmuje wyniki eksperymentów i badań nad wydajnością, a drugi to źródła skryptów budujących środowisko. Praca zawiera niestety dość dużą liczbę drobnych błędów redakcyjnych, ale nie wpływają one w sposób istotny na jej czytelność i wartość. W całej pracy przewijają się samodzielne, zdecydowane wnioski



Autorów, które są wynikiem własnych i oryginalnych badań. Rozdział 5 i dodatki pracy przekonują mnie, że Dyplomanci dość dobrze zrealizowali postawione przed nimi zadanie. Pozwala to stwierdzić, że wykazali się więc także umiejętnością zastosowania w praktyce wiedzy przedstawionej w rozdziałach 2-4. Kończący pracę rozdział szósty świadczy o dużym (ale moim zdaniem uzasadnionym) poczuciu własnej wartości i jest świadectwem własnego, oryginalnego spojrzenia na tematykę przedstawioną w pracy dyplomowej. Uważam, że cele postawione w założeniach pracy zostały pomyślnie zrealizowane. Proponuję ocenę bardzo dobrą (5).

(data, podpis)