

**Politechnika Wrocławskaw  
Wydział Informatyki i Telekomunikacji**

---

Kierunek: **Informatyka techniczna**  
Specjalność: **Inżynieria systemów informatycznych**

**PRACA DYPLOMOWA  
INŻYNIERSKA**

**Aplikacja internetowa do gromadzenia  
i udostępniania informacji o  
technologiach programistycznych**

**A web application for gathering and  
sharing information about software  
development**

Kamil Dywan

Opiekun pracy  
**dr inż, Paweł Rogaliński**

# **Streszczenie**

**Słowa kluczowe:**

## **Abstract**

**Keywords:**

# Spis treści

<b>1. Wstęp . . . . .</b>	<b>8</b>
1.1. Wprowadzenie . . . . .	8
1.2. Cel pracy . . . . .	11
1.3. Zakres pracy . . . . .	11
<b>2. Projektowanie . . . . .</b>	<b>12</b>
2.1. Wymagania funkcjonalne . . . . .	12
2.1.1. Słownik pojęć . . . . .	12
2.1.2. Wymagania funkcjonalne dla systemu IT-Tech . . . . .	15
2.2. Wymagania niefunkcjonalne . . . . .	18
2.3. Architektura systemu . . . . .	18
2.3.1. Baza danych . . . . .	20
2.3.2. Aplikacja serwerowa - Backend . . . . .	20
2.3.3. Klient - Frontend . . . . .	20
2.3.4. Serwer webowy . . . . .	21
2.3.5. Warstwa bezpieczeństwa . . . . .	21
2.4. Diagramy przypadków użycia . . . . .	21
2.4.1. Identyfikacja aktorów . . . . .	21
2.4.2. Diagramy przypadków użycia . . . . .	22
2.4.3. Opisy przypadków użycia . . . . .	27
2.5. Projekt baz danych . . . . .	78
<b>3. Implementacja . . . . .</b>	<b>84</b>
3.1. Technologie i narzędzia . . . . .	84
3.1.1. Technologie . . . . .	84
3.1.2. Narzędzia . . . . .	86
3.2. Aplikacja serwerowa . . . . .	86
3.3. Aplikacja kliencka . . . . .	92
3.4. Warstwa bezpieczeństwa . . . . .	95
<b>4. Testowanie . . . . .</b>	<b>96</b>
<b>5. Podsumowanie . . . . .</b>	<b>97</b>
<b>Literatura . . . . .</b>	<b>98</b>
<b>A. Instrukcja wdrożeniowa . . . . .</b>	<b>99</b>
<b>B. Opis załączonej płyty CD/DVD . . . . .</b>	<b>100</b>

# Spis rysunków

1.1. Wikipedia . . . . .	9
1.2. Medium . . . . .	10
2.1. Cykl życia artykułu . . . . .	14
2.2. Architektura systemu . . . . .	19
2.3. Dziedziczenie między aktorami . . . . .	22
2.4. Diagram przypadków użycia dla każdego użytkownika . . . . .	23
2.5. Diagram przypadków użycia dla niezalogowanego użytkownika . . . . .	23
2.6. Diagram przypadków użycia dla zalogowanego użytkownika . . . . .	24
2.7. Diagram przypadków użycia dla autora artykułu . . . . .	24
2.8. Diagram przypadków użycia dla autora komentarza o artykule . . . . .	25
2.9. Diagram przypadków użycia dla autora opinii o artykule . . . . .	25
2.10. Diagram przypadków użycia dla autora akceptacji opinii artykułu . . . . .	25
2.11. Diagram przypadków użycia dla recenzenta . . . . .	26
2.12. Diagram przypadków użycia dla administratora . . . . .	26
2.13. Diagram przypadków użycia dla systemu . . . . .	27
2.14. Nagłówek strony niezalogowanego użytkownika . . . . .	28
2.15. Formularz rejestracji . . . . .	29
2.16. Formularz służący do akceptacji regulaminu . . . . .	30
2.17. Formularz logowania . . . . .	32
2.18. Nagłówek strony zalogowanego użytkownika . . . . .	32
2.19. Formularz do ustawienia hasła po zresetowaniu hasła . . . . .	32
2.20. Nagłówek strony zalogowanego użytkownika z otwartymi opcjami dotyczącymi konta	33
2.21. Panel z danymi użytkownika . . . . .	34
2.22. Panel umożliwiający użytkownikowi edycję własnych danych . . . . .	35
2.23. Formularz zmiany hasła . . . . .	36
2.24. Nagłówek strony administratora . . . . .	38
2.25. Panel do wyszukiwania użytkowników . . . . .	38
2.26. Panel z danymi użytkownika widziany przez administratora . . . . .	39
2.27. Panel umożliwiający administratorowi edycję danych użytkownika . . . . .	40
2.28. Nagłówek strony dostępnego recenzenta z otwartymi opcjami dotyczącymi konta .	43
2.29. Formularz z ustawieniem daty, do której ma być ustaliona nieobecność recenzenta razem z otwartym oknem wyboru daty . . . . .	43
2.30. Nagłówek strony niedostępnego recenzenta z otwartymi opcjami dotyczącymi konta	43
2.31. Panel do wyszukiwania artykułów z podanymi kryteriami wyszukiwania oraz listą wyszukanych artykułów . . . . .	45
2.32. Dostępne opcje sortowania artykułów . . . . .	46
2.33. Panel z zawartością artykułu . . . . .	47
2.34. Panel z formularzem dodania lub edytowania artykułu . . . . .	49
2.35. Nagłówek strony recenzenta . . . . .	51

---

2.36. Panel z listą artykułów przypisanych recenzentowi do weryfikacji . . . . .	52
2.37. Panel służący do weryfikacji artykułu . . . . .	53
2.38. Formularz akceptacji artykułu z podaną informacją zwrotną . . . . .	54
2.39. Formularz odrzucenia artykułu . . . . .	56
2.40. Panel z zawartością artykułu widziany z perspektywy administratora . . . . .	58
2.41. Panel z listą komentarzy o artykule . . . . .	59
2.42. Formularz dodania lub edytowania komentarza o artykule . . . . .	60
2.43. Dostępne opcje do zarządzania komentarzem . . . . .	61
2.44. Lista komentarzy o artykule widziana z perspektywy administratora . . . . .	62
2.45. Panel z listą opinii o artykule . . . . .	63
2.46. Uzupełniony przykładowymi danymi formularz dodania lub edytowania opinii o artykule . . . . .	64
2.47. Panel z listą opinii o artykule widziany z perspektywy administratora . . . . .	66
2.48. Panel do wyszukiwania technologii z podanymi kryteriami wyszukiwania oraz listą wyszukanych technologii . . . . .	68
2.49. Dostępne opcje sortowania technologii . . . . .	69
2.50. Panel z zawartością technologii . . . . .	70
2.51. Panel z formularzem dodania lub modyfikacji technologii . . . . .	73
2.52. Panel do wyszukiwania technologii z podanymi kryteriami wyszukiwania i listą technologii widziany z perspektywy recenzenta . . . . .	74
2.53. Panel do wyszukiwania technologii, w których recenzent jest ekspertem . . . . .	76
2.54. Formularz przypisania technologii do recenzenta . . . . .	77
2.55. Diagram ERD dla głównej bazy danych . . . . .	78
3.1. Użyte technologie . . . . .	84
3.2. Architektura aplikacji serwerowej . . . . .	86
3.3. Struktura projektu aplikacji serwerowej . . . . .	87
3.4. Model komentarza w aplikacji serwerowej . . . . .	88
3.5. Serwis dla komentarzy w aplikacji serwerowej . . . . .	88
3.6. Kontroler dla komentarzy w aplikacji serwerowej . . . . .	89
3.7. Mapowanie między modelem komentarza, a dto w aplikacji serwerowej . . . . .	89
3.8. Endpointy dla użytkowników, komentarzy oraz artykułów w aplikacji serwerowej . . . . .	90
3.9. Endpointy dla weryfikacji artykułów, dostępności użytkowników oraz technologii, w których specjalizują się recenzenci w aplikacji serwerowej . . . . .	90
3.10. Endpointy dla technologii i kategorii technologii w aplikacji serwerowej . . . . .	91
3.11. Struktura projektu aplikacji klienckiej . . . . .	92
3.12. Model technologii w aplikacji klienckiej . . . . .	93
3.13. Slice zalogowanego użytkownika w aplikacji klienckiej . . . . .	93
3.14. Komponent nawigacja w aplikacji klienckiej . . . . .	94
3.15. Serwis API dla technologii w aplikacji klienckiej . . . . .	94
3.16. Panel administratora Keycloak . . . . .	95

# Spis tabel

2.1. Opis przypadku użycia - rejestracja . . . . .	27
2.2. Opis przypadku użycia - logowanie . . . . .	30
2.3. Opis przypadku użycia - wylogowanie . . . . .	32
2.4. Opis przypadku użycia - edycja własnych danych . . . . .	33
2.5. Opis przypadku użycia - zmiana hasła . . . . .	35
2.6. Opis przypadku użycia - edycja danych innych użytkowników . . . . .	36
2.7. Opis przypadku użycia - zarządzanie rolami użytkownika . . . . .	40
2.8. Opis przypadku użycia - zmiana własnej dostępności . . . . .	41
2.9. Opis przypadku użycia - wyszukiwanie artykułów . . . . .	44
2.10. Opis przypadku użycia - sortowanie artykułów . . . . .	45
2.11. Opis przypadku użycia - wyświetlenie zawartości artykułu . . . . .	46
2.12. Opis przypadku użycia - utworzenie artykułu . . . . .	48
2.13. Opis przypadku użycia - zarządzanie własnym artykułem . . . . .	49
2.14. Opis przypadku użycia - wyświetlenie listy przypisanych artykułów . . . . .	51
2.15. Opis przypadku użycia - weryfikacja artykułu . . . . .	52
2.16. Opis przypadku użycia - akceptacja artykułu . . . . .	53
2.17. Opis przypadku użycia - odrzucenie artykułu . . . . .	54
2.18. Opis przypadku użycia - zarządzanie artykułami . . . . .	56
2.19. Opis przypadku użycia - przeglądanie komentarzy o artykule . . . . .	59
2.20. Opis przypadku użycia - dodanie komentarza o artykule . . . . .	59
2.21. Opis przypadku użycia - zarządzanie własnym komentarzem o artykule . . . . .	60
2.22. Opis przypadku użycia - usuwanie komentarzy o artykule . . . . .	61
2.23. Opis przypadku użycia - przeglądanie opinii o artykule . . . . .	62
2.24. Opis przypadku użycia - dodanie opinii o artykule . . . . .	63
2.25. Opis przypadku użycia - zarządzanie własną opinią o artykule . . . . .	64
2.26. Opis przypadku użycia - usuwanie opinii o artykule . . . . .	65
2.27. Opis przypadku użycia - dodanie akceptacji opinii o artykule . . . . .	66
2.28. Opis przypadku użycia - usunięcie akceptacji opinii o artykule . . . . .	66
2.29. Opis przypadku użycia - wyszukiwanie technologii . . . . .	67
2.30. Opis przypadku użycia - sortowanie technologii . . . . .	68
2.31. Opis przypadku użycia - wyświetlenie opisu technologii . . . . .	69
2.32. Opis przypadku użycia - edycja klasyfikacji technologii . . . . .	70
2.33. Opis przypadku użycia - wyszukiwanie artykułów z przeterminowaną weryfikacją . . . . .	74
2.34. Opis przypadku użycia - przypisanie recenzentowi artykułu do weryfikacji . . . . .	75
2.35. Opis przypadku użycia - przypisanie artykułu do weryfikacji innemu recenzentowi . . . . .	75
2.36. Opis przypadku użycia - wyszukiwanie technologii, w których recenzent jest ekspertem . . . . .	76
2.37. Opis przypadku użycia - zarządzanie listą technologii, w których recenzent jest ekspertem . . . . .	77

# Spis listingów

2.1.	Schemat kolekcji artykułów w postaci JSON dla bazy danych artykułów . . . . .	79
2.2.	Przykładowy dokument dla kolekcji artykułów zgodny ze zdefiniowanym schematem	80
2.3.	Schemat kolekcji komentarzy w postaci JSON dla bazy danych artykułów . . . . .	80
2.4.	Przykładowy dokument dla kolekcji komentarzy zgodny ze zdefiniowanym schematem	81
2.5.	Schemat kolekcji opinii w postaci JSON dla bazy danych artykułów . . . . .	82
2.6.	Przykładowy dokument dla kolekcji opinii zgodny ze zdefiniowanym schematem .	83

# Rozdział 1

## Wstęp

### 1.1. Wprowadzenie

Z powodu szybkiego rozwoju informatyki, jest ciągła potrzeba bycia na bieżąco z technologiami programistycznymi oraz obserwowania trendów. Ważne jest zatem to, aby publikowane treści były jak najbardziej aktualne. Nieaktualne treści najczęściej występują w książkach, gdyż pisanie książki trwa zazwyczaj bardzo długo np. rok i często się zdarza, że książka w dniu wydania już zawiera treści przestarzałe. Aplikacja, której dotyczy ta praca będzie umożliwiała szybkie zamieszczanie treści każdemu użytkownikowi i dzięki temu, możliwe będzie dostarczanie treści aktualnych i nadążanie za trendami.

Programista, który chciałby się czegoś dowiedzieć o technologii, z którą nie jest zaznajomiony, najczęściej może skorzystać z dokumentacji tej technologii albo serwisu informacyjnego dostarczającego treści z wielu dziedzin (np. Wikipedia). Problemem jest tutaj to, że nie ma popularnych serwisów informacyjnych, które zawierają treści będące pomiędzy dokumentacjami i ogólnymi serwisami informacyjnymi. Z jednej strony dokumentacje technologii zawierają szczegółowe treści oraz rzadko kiedy wplatane są jakieś przykłady, a z drugiej strony encyklopedyczne serwisy informacyjne zawierają jedynie treści ogólne. Dokumentacja może być niezrozumiała dla tego użytkownika, a serwisy informacyjne obejmujące wiele dziedzin najczęściej nie dostarczą mu wystarczających informacji. Aplikacja, której dotyczy ta praca jest rozwiązaniem pośrednim, gdyż nie będzie nacisku na dużą szczegółowość zamieszczanych treści, jak i z powodu tego, że tematyka serwisu to technologie programistyczne, to treści nie będą zbyt ogólne.

Aktualnie nie ma popularnych serwisów informacyjnych zorientowanych jedynie na technologiach programistycznych, ale istnieje na rynku dużo serwisów, które są zorientowane na wiele dziedzin.

Jednym z takich serwisów jest niezwykle popularna Wikipedia 1.1. Wikipedia jest to serwis obejmujący bardzo wiele dziedzin, który umożliwia użytkownikom publikowanie treści w postaci artykułów. Treści publikowane w tym serwisie zazwyczaj są utrzymane w tonie encyklopedii. Serwis ten jest zorientowany na informacje i dlatego posiada słabo rozwiniętą formę społecznościową. Przy każdym z artykułów jest sekcja „Dyskusja”, w której można umieścić komentarz lub odpowiedzieć na komentarz innego użytkownika.

The screenshot shows the Polish Wikipedia page for the programming language Java. At the top, there's a navigation bar with links for 'Artykuł', 'Dyskusja', 'Czytaj', 'Edytuj', 'Edytuj kod źródłowy', 'Historia i autorzy', 'Przeszukaj Wikipedię', and a search icon. Below the title 'Java [edytuj]' is a note: 'Ten artykuł dotyczy języka programowania. Zobacz też: inne znaczenia nazwy „Java”.' The main content area starts with a brief introduction: 'Java (wym. dżawa<sup>[4]</sup>) – wspólnieznny, oparty na klasach, obiektowy język programowania ogólnego zastosowania<sup>[5]</sup>. Został stworzony przez grupę roboczą pod kierunkiem Jamesa Goslinga z firmy Sun Microsystems. Java jest językiem tworzenia programów źródłowych komplikowanych do kodu bajtowego, czyli postaci wykonywanej przez maszynę wirtualną. Język cechuje się silnym typowaniem. Jego podstawowe koncepcje zostały przejęte z języka Smalltalk (maszyna wirtualna, zarządzanie pamięcią) oraz z języka C++ (duża część składni i słów kluczowych).'. To the right is a sidebar with a large Java logo (a stylized character with a red eye), and sections for 'Pojawienie się 1995' and 'Paradygmat Wieloparadygmatowy (obiektowy, strukturalny, imperatywny, funkcyjny)'. A table of contents on the left lists sections like 'Historia', 'Główne koncepcje', 'Obiektowość', etc.

Rys. 1.1: Wikipedia

Źródło: <https://pl.wikipedia.org/wiki/Java>

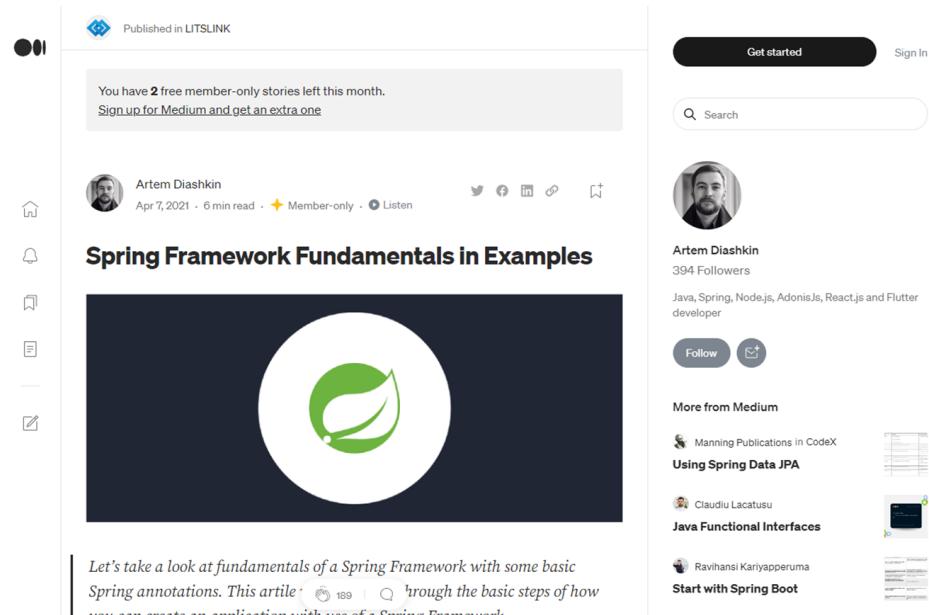
Można wyróżnić następujące zalety Wikipedii:

- czytelne i ładne artykuły,
- artykuły posiadają podobną budowę,
- darmowy dostęp dla każdego użytkownika,
- rozbudowany formularz tworzenia i edycji artykułu.

Do wad Wikipedii można zaliczyć:

- sekcja komentarzy jest nieczytelna,
- za mało kryteriów wyszukiwania,
- często artykuły są zbyt ogólne,
- brak systemu opinii.

Podobnym serwisem do Wikipedii jest Medium 1.2. Medium również obejmuje wiele dziedzin. Użytkownicy także publikują treści w tym serwisie poprzez artykuły, jednak treści te są najczęściej utrzymane w swobodniejszym tonie, niż encyklopedycznym. Serwis ten można zaliczyć do serwisów społecznościowych. Na początku każdego artykułu widnieje wyraźna informacja o autorze, można wejść na profil tego autora, zaobserwować go albo zobaczyć napisane przez niego artykuły. W serwisie Medium jest również system zagnieżdżonych komentarzy oraz oceniania artykułów. Każdy użytkownik może zostawić jednostopniową pochwałę do artykułu lub komentarza.



Rys. 1.2: Medium

Źródło: <https://medium.com/litslink/spring-framework-fundamentals-in-examples-b37101aa9250>

Można wyróżnić następujące zalety Medium:

- czytelne i ładne artykuły,
- rozbudowany system wyszukiwania artykułów (m.in. po kategorii, czy autorze),
- czytelna sekcja z komentarzami,
- rozbudowana forma społecznościowa.

Do wad Medium można zaliczyć:

- budowa artykułów może się mocno różnić,
- zbyt prosty formularz do tworzenia artykułów,
- zbyt mały zakres zostawianych ocen (jedynie jednostopniowa pochwała),
- aby mieć dostęp do wszystkich artykułów, trzeba posiadać płatne konto premium,
- brak obowiązkowego tekstowego uzasadnienia oceny artykułu.

Systemy umożliwiające dostarczanie treści każdemu użytkownikowi są szczególnie narażone na to, że publikowane treści będą nieodpowiednie np. spam, wirusy, czy informacje nieprawdziwe. W tego typu serwisach niezbędna jest moderacja. Niestety, ale w przypadku popularnych serwisów tego typu, często niemożliwe jest zapewnienie odpowiedniej liczby moderatorów i potrzebny jest wtenczas mechanizm wspomagający moderatorów. Mechanizmem tym może być system komentarzy i opinii. Informacje zwrotne użytkowników na temat treści dostępnych w serwisie będą mogły być wtedy informacją o jakości tych treści zarówno dla moderatorów, jak i innych użytkowników. W przypadku wielu serwisów informacyjnych mechanizm zostawiania informacji zwrotnej (komentarze, oceny, opinie) jest słaby lub niewystarczająco rozbudowany. Najczęściej można się spotkać z następującymi niedociągnięciami:

- sekcja komentarzy jest nieczytelna oraz nie można odpowiedzieć na komentarz innego użytkownika,
- w przypadku opinii jest zbyt mały zakres wystawianej oceny (np. łapka w góre i w dół albo sama pochwała), jak i często nie ma wymogu zamieszczenia tekstu uzasadnienia do wystawionej oceny,
- brak możliwości ocenienia opinii innego użytkownika.

## **1.2. Cel pracy**

Celem pracy jest zaprojektowanie i implementacja responsywnej aplikacji webowej IT-Tech, która będzie umożliwiała użytkownikom tworzenie artykułów o technologiach programistycznych (np. dotyczących języków programowania), przeglądanie dostępnych artykułów oraz zostawienie informacji zwrotnej o artykule w postaci komentarza lub opinii.

## **1.3. Zakres pracy**

Zakres pracy obejmuje: projekt, implementację, testowanie oraz wdrożenie i opracowanie dokumentacji. Projekt obejmuje: zdefiniowanie wymagań funkcjonalnych i niefunkcjonalnych, sporządzanie diagramów i opisów przypadków użycia, zrobienie widoków interfejsu użytkownika oraz wykonanie schematów bazy danych. Implementacja obejmuje: aplikację serwerową (backend), aplikację kliencką (frontend) oraz uwierzytelnianie i autoryzację. Testowanie obejmuje: testy integracyjne backendu oraz testy manualne i automatyczne interfejsu użytkownika.

## Rozdział 2

# Projektowanie

### 2.1. Wymagania funkcjonalne

Pierwszym etapem realizacji pracy jest określenie wymagań funkcjonalnych i niefunkcjonalnych. W tym celu konieczne jest wprowadzenie pojęć, które będą wykorzystywane w dalszej części pracy.

#### 2.1.1. Słownik pojęć

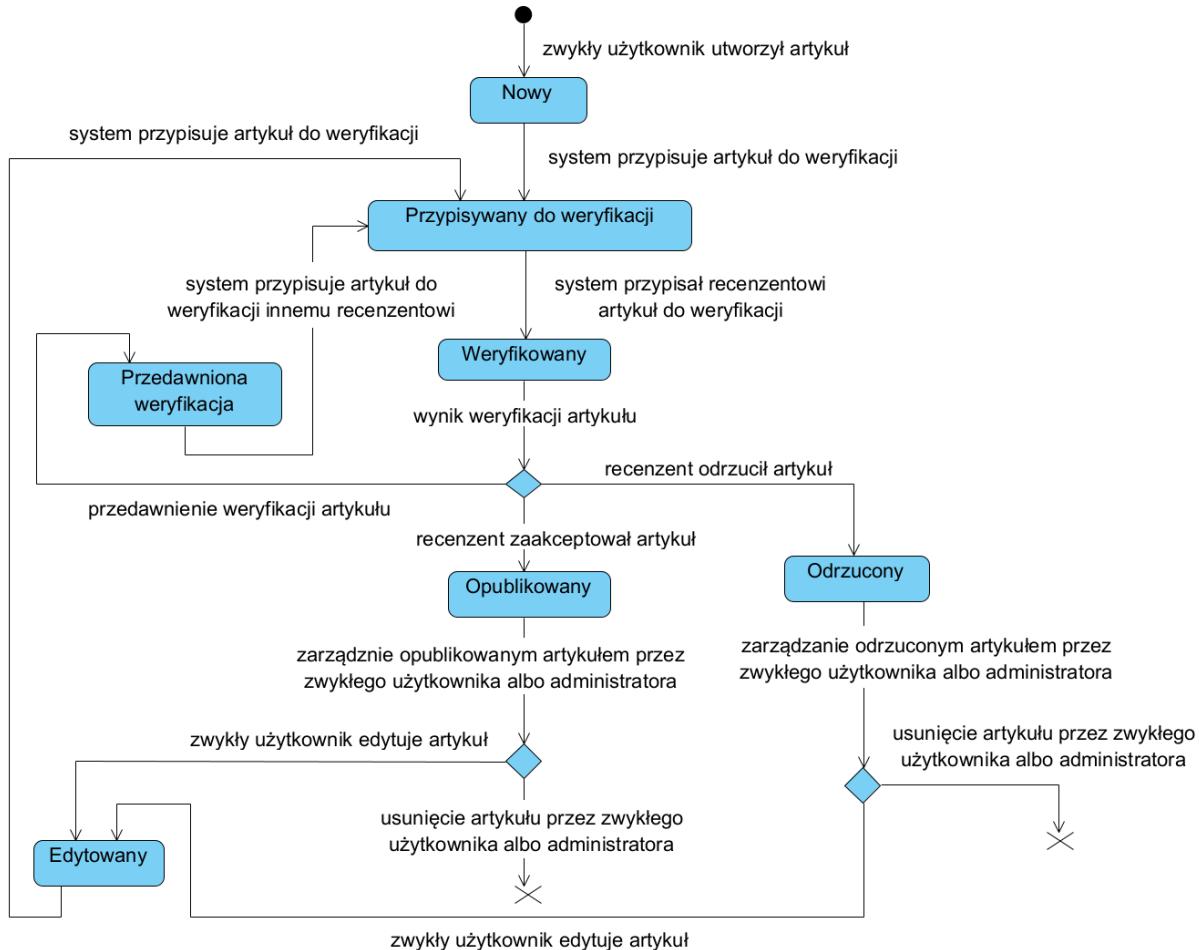
Przed zdefiniowaniem wymagań funkcjonalnych, potrzebne było określenie pojęć, które będą używane zarówno przy wymaganiach funkcjonalnych, jak i w reszcie pracy.

W pracy będą wykorzystywane następujące pojęcia:

- **Technologia** - pakiet oprogramowania lub narzędzia developerskie wykorzystywane przez informatyków, a w szczególności przez programistów. Technologia będzie opisywana następującymi atrybutami:
  - Nazwa,
  - Opis,
  - Dostawca (opcjonalne),
  - Ikona (opcjonalne),
  - Kategoria,
  - Data pierwszego wydania (opcjonalne),
  - Data ostatniego wydania (opcjonalne),
  - Daty edycji.
- **Podział technologii** - hierarchiczny podział technologii na kategorie ze względu na ich architekturę. Klasyfikację tę można przedstawić w postaci następującego drzewa:
  - Technologie informatyczne
    - \* Język programowania (np. Java, C++)
      - Biblioteka (np. SFML, SDL, OpenGL),
      - Framework (np. Spring, Spring Boot, React, Angular)
    - \* Język znaczników (np. TeX, HTML, XML)
    - \* Język bazy danych (np. SQL, GraphQL)
      - Relacyjna (np. MySQL, Oracle Database, SQLite)
      - Obiektowo-relacyjna (np. PostgreSQL)
      - NoSQL (np. MongoDB, Cassandra)

- Środowiska uruchomieniowe
  - \* System operacyjny (np. Windows, Linux)
  - \* Środowiska wirtualizowane (JVM, .NET, node.js)
  - \* Systemy wbudowane (np. Arduino, Raspberry Pi)
- Narzędzia developerskie
  - \* Środowisko programistyczne (np. Netbeans, Eclipse, Visual Studio Code, Visual Studio, ItelliJ)
  - \* System kontroli wersji (np. git)
    - Serwisy hostujące gita (np. GitLab, GitHub)
  - \* CI/CD - ciągła integracja/ciągłe dostarczanie (np. Jenkins, GitLab CI)
  - \* Konteneryzacja (np. docker)
  - \* Orkiestracja (system do zarządzania, organizacji i planowania zasobów systemu – np. Docker Compose, Kubernetes)
- Inne (oprogramowanie, które nie zostało przypisane do żadnej z ww. kategorii)
- Artykuł - opis danej technologii przygotowany przez użytkownika systemu. Artykuł będzie opisywany następującymi atrybutami:
  - Status (możliwe wartości: nowy, przypisywany do weryfikacji, weryfikowany, przedawiona weryfikacja, odrzucony, opublikowany, edytowany),
  - Tytuł,
  - Autor,
  - Data utworzenia,
  - Data modyfikacji,
  - Kategoria technologii wraz z całą hierarchiczną ścieżką prowadzącą do tej kategorii (np. dla Javy będzie to: Technologie Informatyczne / Język Programowania / Java),
  - Dostawca technologii (opcjonalne),
  - Ikona technologii (opcjonalne),
  - Tekst artykułu,
  - Lista opinii o artykule,
  - Średnia ocena z co najmniej 5-ciu opinii o artykule,
  - Data przypisania artykułu do weryfikacji,
  - Recenzent przypisany do weryfikacji artykułu.

Artykuły podlegają cyklowi życia, który jest przedstawiony na rysunku 2.1:



Rys. 2.1: Cykl życia artykułu

Źródło: opracowanie własne

Na rysunku 2.1 zostały przedstawione przejścia do odpowiednich statusów artykułu (zaokrąglone prostokąty) w wyniku podjętych przez użytkowników działań.

- **Weryfikacja artykułu** - informacja o tym, czy artykuł powinien zostać opublikowany w serwisie (zaakceptowanie artykułu) albo wymaga jeszcze korekty (odrzucenie artykułu),
- **Dostępność recenzenta** - informacja o tym, czy recenzent może weryfikować artykuły w danym czasie,
- **Komentarz do artykułu** - tekstowa informacja zwrotna zalogowanego użytkownika o danym artykule,
- **Ocena artykułu** - ocena wystawiona przez zalogowanego użytkownika na temat danego artykułu. Ocena ta jest w postaci gwiazdek będąca wartością od 1 do, przy czym wartość 1 oznacza ocenę negatywną i wskazuje, że artykuł posiada rażące błędy i powinien zostać edytowany albo usunięty, a wartość 5 oznacza ocenę pozytywną i wskazuje, że artykuł nie posiada rażących błędów oraz informacje w nim zawarte są niezwykle przydatne dla użytkowników,
- **Opinia o artykule** - oceniająca informacja zwrotna zalogowanego użytkownika na temat danego artykułu, która zawiera tekstowy komentarz oraz ocenę tego artykułu,
- **Akceptacja opinii o artykule** - ocena wystawiona przez zalogowanego użytkownika dotycząca opinii o artykule. Przyjmuje ona dwie wartości: ocena pozytywna (łapka w góre) oznaczająca, że użytkownik zgadza się z opinią i ocena negatywna (łapka w dół) oznaczającą, iż użytkownik nie zgadza się z opinią,
- **Rola** - zestaw uprawnień użytkownika, które określają jego zakres dostępu do systemu oraz określonych funkcji tego systemu. Możliwe jest przypisanie danemu użytkownikowi ról, co

jest równoznaczne z uzyskaniem przez tego użytkownika uprawnień zdefiniowanych w tych rolach. Wyróżniono w systemie następujące role:

- **Użytkownik** - każdy użytkownik korzystający z serwisu. Może on jedynie przeglądać zasoby serwisu,
- **Niezalogowany użytkownik** - niezalogowany użytkownik mający możliwość przeglądania zasobów serwisu oraz może się zarejestrować i zalogować,
- **Zalogowany użytkownik** - każdy zarejestrowany i zalogowany użytkownik. Może on przede wszystkim przeglądać i tworzyć artykuły oraz wystawiać komentarze i opinie o artykułach.
- **Autor artykułu** - zalogowany użytkownik mogący edytować oraz usuwać utworzone przez siebie artykuły,
- **Autor komentarza o artykule** - zalogowany użytkownik mogący edytować oraz usuwać utworzone przez siebie komentarze o artykule,
- **Autor opinii o artykule** - zalogowany użytkownik mogący edytować oraz usunąć utworzoną przez siebie opinię o artykule,
- **Autor akceptacji opinii o artykule** - zalogowany użytkownik mogący usunąć utworzoną przez siebie akceptację opinii o artykule,
- **Recenzent (ekspert od technologii)** - zalogowany użytkownik, który otrzymał od administratora rolę recenzenta. Jego głównym zadaniem jest weryfikacja artykułów przed ich opublikowaniem w serwisie. Innym uprawnieniem tego użytkownika jest m.in. zarządzanie klasyfikacją technologii.
- **Administrator (utrzymuje porządek w serwisie)** - zalogowany użytkownik, który otrzymał od innego administratora rolę administratora. Odpowiada on przede wszystkim za przydzielanie ról użytkownikom oraz utrzymywanie porządku w serwisie poprzez edytowanie i usuwanie niewłaściwych treści (np. wirusy lub spam). Przy instalacji systemu powinno być tworzone co najmniej jedno konto użytkownika z rolą administratora, gdyż to właśnie administrator odpowiada za przydzielanie ról innym użytkownikom,
- **System** - wbudowane procedury, które umożliwiają automatyczne przypisywanie recenzenta do weryfikacji artykułów oraz automatyczne ponowne przypisanie artykułu do weryfikacji innemu recenzentowi, jeśli upłynął 1-tygodniowy termin weryfikacji artykułu przez aktualnego recenzenta.

## 2.1.2. Wymagania funkcjonalne dla systemu IT-Tech

Po zdefiniowaniu pojęć, przystąpiono do określenia wymagań funkcjonalnych dla poszczególnych użytkowników.

Zdefiniowano następujące wymagania funkcjonalne:

- **Użytkownik** powinien mieć możliwość:

- wyszukiwania artykułów po następujących kryteriach:
  - \* Tytuł,
  - \* Autor,
  - \* Kategoria technologii,
  - \* Technologia,
  - \* Dostawca technologii
  - \* Zakres dat, w których mieści się data powstania artykułu,
  - \* Zakres dat, w których mieści się data modyfikacji artykułu.
- sortowania artykułów według następujących kryteriów:

- \* Nazwa,
  - \* Nazwa technologii,
  - \* Autor,
  - \* Data powstania,
  - \* Data modyfikacji,
  - \* Średnia ocena,
  - \* Liczba opinii.
- wyświetlenia zawartości artykułu,
  - przeglądania komentarzy o artykule,
  - przeglądania opinii o artykule,
  - wyszukiwania technologii po następujących kryteriach:
    - \* Nazwa,
    - \* Kategoria,
    - \* Dostawca,
    - \* Zakres dat, w których mieści się data pierwszego wydania,
    - \* Zakres dat, w których mieści się data ostatniego wydania.
  - sortowania technologii według następujących kryteriów:
    - \* Nazwa,
    - \* Kategoria,
    - \* Data pierwszego wydania,
    - \* Data ostatniego wydania.
  - wyświetlenia opisu technologii,
- Niezalogowany użytkownik powinien mieć możliwość:
    - zarejestrowania się
      - \* Podczas rejestracji użytkownik będzie podawał następujące informacje:
        - Nazwa użytkownika,
        - Pseudonim (nazwa, pod którą użytkownik będzie mógł być rozpoznany w serwisie przez innych użytkowników)
        - Imię,
        - Nazwisko,
        - Adres e-mail,
        - (Opcjonalnie) Avatar,
        - Hasło.
      - \* Po rejestracji i zalogowaniu, użytkownik uzyskuje uprawnienia zalogowanego użytkownika, a aby uzyskać uprawnienia recenzenta lub administratora, należy w tym celu skontaktować się z administratorem, który będzie w stanie nadać taką rolę,
    - zalogowania się
      - \* Podczas logowania użytkownik będzie wpisywał login albo e-mail oraz hasło.
  - Zalogowany użytkownik powinien mieć możliwość:
    - edycji własnych danych,
    - zmiany hasła,
    - wylogowania się,
    - stworzenia artykułu, przy czym aby artykuł został opublikowany w serwisie, to artykuł ten musi przejść pozytywnie weryfikację przeprowadzaną przez recenzenta,
    - dodania komentarza o artykule,
    - dodania jednej opinii o artykule którego użytkownik nie jest autorem,

- dodania jednej akceptacji opinii o artykule wystawionej przez innego użytkownika.
- **Autor artykułu** powinien mieć możliwość:
  - zarządzania własnymi artykułami w następującym zakresie:
    - \* edytowanie, przy czym aby zmieniony artykuł został opublikowany w serwisie, to artykuł ten musi przejść pozytywnie weryfikację przeprowadzaną przez recenzenta,
    - \* usuwanie.
- **Autor komentarza o artykule** powinien mieć możliwość:
  - zarządzania własnymi komentarzami o artykule w następującym zakresie:
    - \* edytowanie,
    - \* usuwanie.
- **Autor opinii o artykule** powinien mieć możliwość:
  - zarządzania własną opinią o artykule w następującym zakresie:
    - \* edytowanie,
    - \* usuwanie.
- **Autor akceptacji opinii o artykule** powinien mieć możliwość:
  - usunięcia własnej akceptacji opinii o artykule.
- **Recenzent** powinien mieć możliwość:
  - edycji rodziny technologii,
  - ustawienia własnej dostępności,
  - weryfikacji artykułów według następujących założeń:
    - \* po przypisaniu artykułu do weryfikacji danemu recenzentowi, recenzent ten będzie miał tydzień na akceptacje lub odrzucenie artykułu,
    - \* przy akceptacji artykułu opcjonalne jest załączenie wiadomości (np. drobne uwagi), ale przy odrzuceniu artykułu załączenie wiadomości jest obowiązkowe i powinna ona zawierać informacje o tym, dlaczego artykuł został odrzucony
    - \* jeśli recenzent nie zweryfikuje artykułu w ciągu tygodnia, wtenczas artykuł ten zostanie przypisany do weryfikacji innemu recenzentowi.
- **Administrator** powinien mieć możliwość:
  - edycji danych innych użytkowników,F
  - zarządzania artykułami w następującym zakresie:
    - \* edytowanie,
    - \* usuwanie.
  - usuwania komentarzy o artykule,
  - usuwania opinii o artykule,
  - przypisywania innym użytkownikom ról,
  - usuwania przypisanych ról innych użytkowników.
- **System** powinien mieć co 24 godziny możliwość automatycznego:
  - przypisywania artykułów do weryfikacji dostępnym recenzentom, którzy w ciągu ostatnich 30 dni zrecenzowali najmniejszą liczbę artykułów,
  - wyszukiwania artykułów z przedawnioną weryfikacją,
  - usunięcia przedawnionych weryfikacji artykułów oraz przypisania tych artykułów do ponownej weryfikacji.

## 2.2. Wymagania niefunkcjonalne

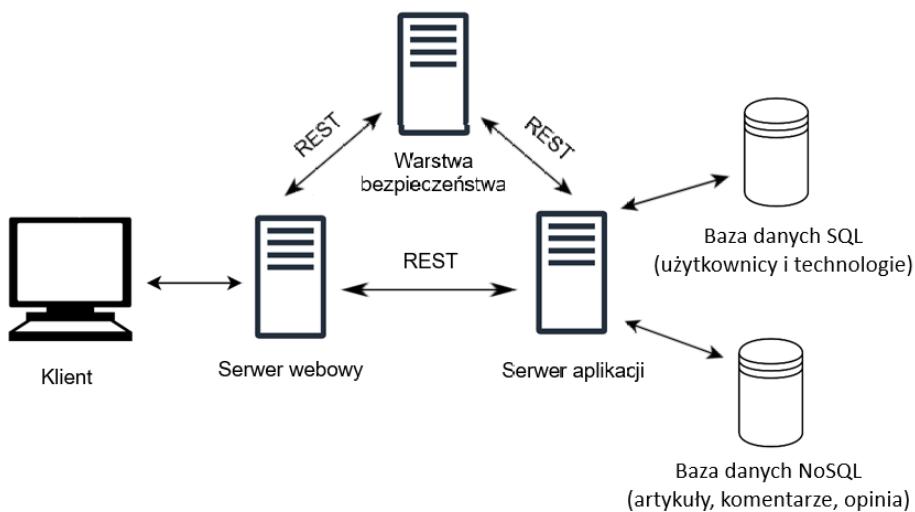
Na podstawie celu oraz zakresu pracy zdefiniowano następujące wymagania niefunkcjonalne:

- Wymagania dotyczące bezpieczeństwa systemu
  - Użytkownik o posiadanej przez siebie roli będzie miał dostęp do systemu jedynie w zakresie uprawnień przypisanych do tej roli,
  - Używanie zewnętrznego serwera uwierzytelniającego i autoryzacyjnego,
  - Używanie tokenu na okaziciela (ang. *Bearer token*) przy wykonywaniu zapytań do Backendu,
  - Nieprzesyłanie hasła, jeśli nie jest to konieczne,
  - Tworzone hasła powinny spełniać następujące kryteria:
    - \* Minimalna długość - 8,
    - \* Co najmniej jedna mała litera,
    - \* Co najmniej jedna duża litera,
    - \* Co najmniej jedna cyfra.
- Wykorzystywane technologie i narzędzia
  - Backend - Spring Boot,
  - Frontend - React (główny framework), Typescript (statyczne typowanie), MUI (biblioteka komponentów),
  - Główna baza danych - PostgreSQL,
  - Baza danych artykułów - MongoDB,
  - Warstwa bezpieczeństwa (serwer uwierzytelniania i autoryzacji) - Keycloak,
  - Dokumentacja - LaTeX.

## 2.3. Architektura systemu

Na podstawie analizy wymagań funkcjonalnych i niefunkcjonalnych, opracowano architekturę systemu. Realizowany system jest rozproszonym serwisem webowym, który w dużym uogólnieniu można opisać jako system typu klient-serwer. Klient wysyła żądanie do serwera aplikacji, a następnie serwer odpowiednio przetwarza otrzymane żądanie i zwraca klientowi odpowiedź, którą klient interpretuje i przedstawia użytkownikowi. Zdecydowano się na rozproszoną architekturę systemu, gdyż taka architektura zapewnia większą niezawodność i skalowalność niż w przypadku systemów scentralizowanych.

Szczegółową architekturę systemu przedstawiono na rysunku 2.2



Rys. 2.2: Architektura systemu

Źródło: opracowanie własne

Zdecydowano się na zaimplementowanie systemu w postaci serwisu webowego głównie dla tego, że taka forma aplikacji zapewnia dużą wygodę użytkowania z powodu braku konieczności jej zainstalowania przez użytkownika. Innym powodem jest to, że serwisy webowe są obecnie najpopularniejszym typem systemu i implementacja takiej aplikacji pozwoli na zdobycie wartościowego doświadczenia.

Komunikacja w systemie przebiega głównie poprzez protokół HTTP (ang. *Hypertext Transfer Protocol*). Protokół ten jest oparty na komunikacji typu klient-serwer, przy czym klientem jest najczęściej przeglądarka. Klientem wysyła żądanie do serwera, a serwer wysyła do klienta odpowiedź. Zawartość wiadomości HTTP można przede wszystkim podzielić na nagłówek (ang. *Header*) oraz ładunek (ang. *Body*). W nagłówku znajdują się pola typu klucz-wartość. Nagłówek żądania różni się od nagłówka odpowiedzi poprzez dostępne opcje, ale mają one również wspólne pola np. informacja o hoście docelowym, czy format wiadomości. W nagłówku żądania można wyróżnić jeszcze np. nagłówek **Authorization**, w którym umieszczane są poświadczenie uwierzytelnienia i autoryzacji. W nagłówku odpowiedzi można za to wyróżnić m.in. status odpowiedzi, który np. informuje o tym, czy żądanie zostało wykonane pomyślnie (np. status 200), czy też wystąpił błąd, a jeśli wystąpił, to z jakiego powodu (np. status 401 oznaczający nieudane uwierzytelnienie użytkownika). Ładunek jest opcjonalnym polem, które stanowi treść wiadomości HTTP i np. w przypadku odpowiedzi może zawierać listę wyszukanych artykułów.

Komunikacja w systemie jest zgodna ze stylem architektury oprogramowania REST (ang. *Representational state transfer*). REST jest to sposób i format w jaki komunikuje się klient z serwerem. Serwer udostępnia klientowi punkty końcowe (end-pointy), do których klient może wysłać żądania HTTP przesyłając przy tym jakieś dane np. tytuł wyszukiwanego artykułu. W skrócie komunikacja REST odznacza się następującymi cechami:

- bezstanowość (nie przechowuje informacji o poprzednich wymianach wiadomości),
- architektura klient-serwer,
- jednolity interfejs komunikacyjny – dzięki temu możliwe jest np. komunikowanie się systemów zaimplementowanych w różnych językach programowania, czy zainstalowanych na różnych platformach (np. mobilna),
- wykorzystywanie protokołu HTTP.

### 2.3.1. Baza danych

Baza danych to warstwa systemu odpowiedzialna za przechowywanie danych. Założono, że w systemie zostaną wykorzystane dwie bazy danych różnego typu: obiektowo-relacyjna (SQL) oraz nierelacyjna oparta na dokumentach (NoSQL).

Jako główną bazę danych wybrano bazę obiektowo-relacyjną, gdyż bazy relacyjne charakteryzują się tym, że zapewniają integralność danych poprzez tworzenie relacji np. poprzez powiązanie tabel kluczem obcym, a w realizowanym systemie wykryto wiele przypadków, w których można utworzyć relacje. Przykładowo właściciel artykułu jest powiązaniem między encjami użytkownika i artykułu oraz historia modyfikacji technologii jest dołączona do danej technologii. Zdecydowano się na obiektową warstwę bazy danych, gdyż upraszcza ona pracę z kilkoma wykorzystywanyimi narzędziami (np. framework Hibernate języka Java), które są oparte na programowaniu obiektowym.

Artykuły mają złożoną zagnieźdzoną strukturę, zawierającą rozdziały, podrozdziały, komentarze oraz opinie. Nierelacyjna baza dokumentowa wydaje się być idealnym pomysłem do przechowywania takich informacji. Baza ta pozwala na przechowywanie struktur typu klucz-wartość, co zapewnia dużą elastyczność w przechowywaniu danych, gdyż pozwala na m.in. przechowywanie zagnieźdzonych struktur oraz umożliwia pobranie danych w postaci zagnieźdzonej struktury nawet poprzez jedno zapytanie. Dla porównania w przypadku relacyjnej bazy danych należałoby wykonać wiele klauzuli JOIN aby pobrać te same dane i przez co traciłoby się na efektywności.

### 2.3.2. Aplikacja serwerowa - Backend

Backend jest odpowiedzialny za przyjmowanie żądań od klienta, odpowiednie przetwarzanie tych żądań, wykonywanie pewnych operacji na danych przechowywanych w bazie danych na podstawie otrzymanych przez klienta danych i przekazywanie klientowi adekwatnej odpowiedzi. Backend udostępnia klientowi punkty końcowe (ang. *API endpoints*) do których klient może przesyłać żądanie. Punkty końcowe posiadają adresy URL np. <http://localhost:9000/user>. Adresy URL składają się nazwy protokołu, nazwy hosta docelowego, portu docelowego oraz reszty, która identyfikuje punkt końcowy oraz jego przeznaczenie. Ważnym elementem w przypadku backendu jest również wysłanie klientowi odpowiedniego statusu odpowiedzi oraz ewentualne dołączenie informacji zwrotnej. Dzięki temu klient będzie mógł odpowiednio zareagować na otrzymaną odpowiedź (np. w przypadku otrzymania negatywnej odpowiedzi, klient wyświetli użytkownikowi komunikat o błędzie, a w przypadku otrzymania odpowiedzi negatywnej, klient wyświetli komunikat o sukcesie). Punkty końcowe backendu są chronione przez warstwę bezpieczeństwa. Klient będzie mógł wysłać żądanie na dany endpoint tylko wtedy, gdy użytkownik po stronie klienta będzie miał wymagane przez ten endpoint uprawnienia (role). Warstwa ta jest ściśle powiązana z warstwą bazy danych.

### 2.3.3. Klient - Frontend

Frontend jest odpowiedzialny za wysyłanie żądań do warstwy backendowej i następnie odpowiednie przetwarzanie oraz wyświetlanie danych otrzymanych w odpowiedzi od backendu. W systemie tym klientem jest strona internetowa renderowana po stronie użytkownika. Klient jest aplikacją typu SPA (ang. *Single Page Application*), co oznacza, że aplikacja składa się z jednej strony HTML, a jej zawartość jest zmieniana dynamicznie poprzez JavaScript. Zdecydowano się na aplikację typu SPA, a nie MPA (ang. *Multi Page Application*) głównie dlatego, że aplikacje SPA działają szybciej z powodu tylko jednej strony HTML. Kolejnymi kryteriami na korzyść SPA były łatwiejsze wdrożenie, gdyż wystarczą do tego jedynie 3 pliki (HTML, CSS i JavaScript), łatwiejsze wprowadzenie dynamicznej zawartości oraz ułatwione tworzenie aplikacji.

Frontend jest również chroniony przez warstwę bezpieczeństwa. Użytkownik będzie miał dostęp do podstron oraz funkcji systemu jedynie w zakresie posiadanych przez niego uprawnień (ról).

### 2.3.4. Serwer webowy

Warstwa odpowiedzialna za dostarczenie treści strony WWW w odpowiedzi na żądania HTTP użytkownika. Użytkownik jest połączony z tym serwerem poprzez swoją przeglądarkę. Innymi zadaniami tej warstwy są przekierowywanie żądań oraz zapewnienie bezpieczeństwa poprzez m.in. możliwość określenia dozwolonych nagłówków w wysyłanych do tego serwera żądaniach.

### 2.3.5. Warstwa bezpieczeństwa

Warstwa bezpieczeństwa jest osobnym serwerem, który odpowiada za uwierzytelnienie i autoryzację. Warstwa ta zabezpiecza zarówno warstwę kliencką, jak i serwer aplikacji. Warstwa kliencka jest chroniona poprzez blokowanie niektórych podstron, jak i funkcji systemu, które nie powinny być dostępne dla danego użytkownika. Mechanizmem bezpieczeństwa dla aplikacji serwerowej jest wymóg dostarczania prawidłowego tokenu dostępu (ang. *Access token*) oraz posiadania odpowiednich uprawnień przy wysyłaniu przez klienta żądania do zabezpieczonego punktu końcowego tego serwera.

Klient może uzyskać token dostępu poprzez zalogowanie się do serwera bezpieczeństwa. Logowanie może być wykonane poprzez wysłanie przez klienta żądania z danymi do logowania na odpowiedni endpoint serwera bezpieczeństwa. W przypadku pomyślnego logowania, klient otrzymuje od serwera bezpieczeństwa m.in. wygenerowane tokeny dostępu oraz odświeżania wraz z datami ważności tych tokenów. Token dostępu jest później dostarczany przez klienta w nagłówku wiadomości w polu przeznaczonym do autoryzacji: *Authorization: Bearer <token>*. Przed wysłaniem żądania do backendu klient sprawdza, czy token dostępu stracił swoją ważność (zwykle kilka minut) i jeśli token ten stracił swoją ważność, to klient wysyła żądanie z dołączonym tokenem odświeżania o wygenerowanie nowego tokenu dostępu, do którego dołączony będzie również aktualny token odświeżania. Token odświeżania może stracić swoją ważność jedynie w przypadku wylogowania albo nieaktywności użytkownika. Opisany proces uwierzytelnienia nazywa się uwierzytelnianiem na okaziciela (ang. *Bearer authentication*). Zastosowany serwer bezpieczeństwa przeprowadza autoryzację użytkowników w oparciu o ich role.

## 2.4. Diagramy przypadków użycia

Na podstawie zdefiniowanych wcześniej wymagań funkcjonalnych, wykonano identyfikację aktorów oraz sporządzono diagramy przypadków użycia. Do stworzenia diagramów wykorzystano program Visual Paradigm Standard.

### 2.4.1. Identyfikacja aktorów

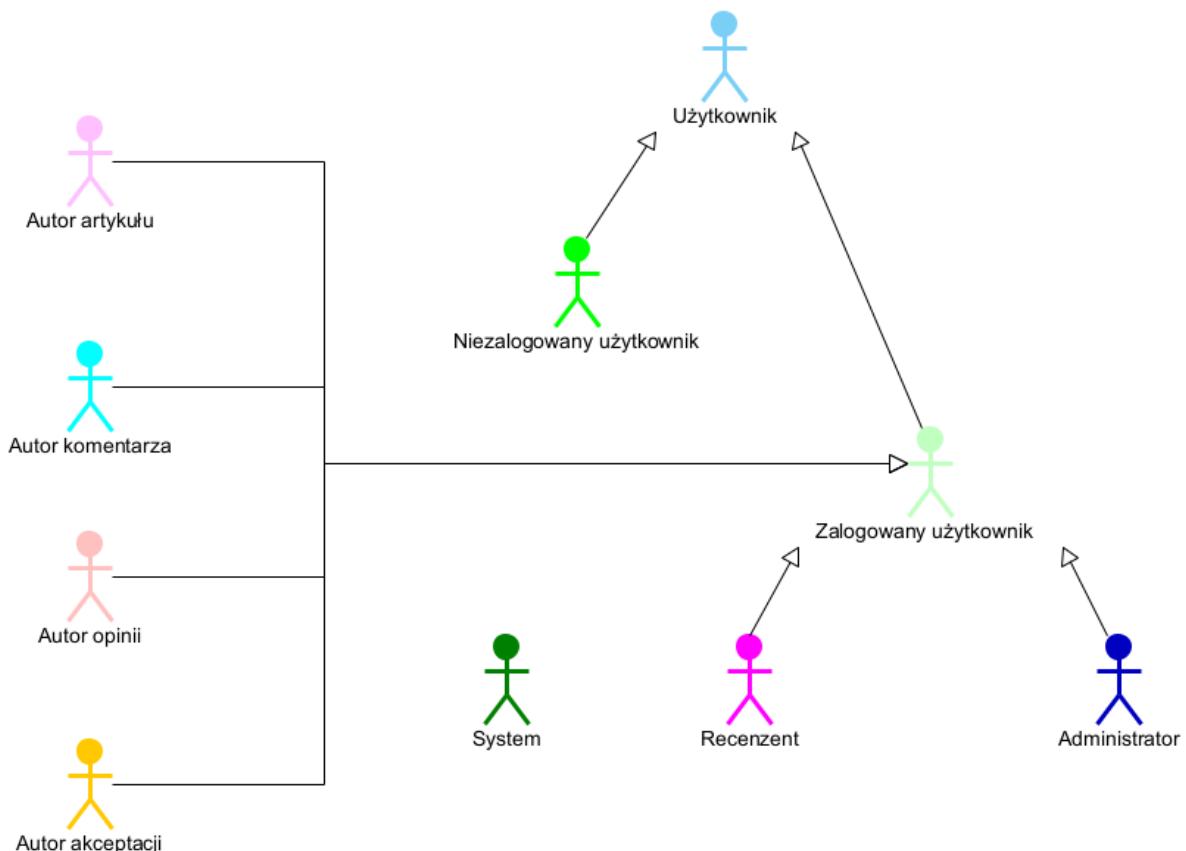
Na podstawie analizy wymagań funkcjonalnych zdefiniowano następujących aktorów:

- Użytkownik,
- Niezalogowany użytkownik,
- Zalogowany użytkownik,
- Autor artykułu,
- Autor komentarza o artykule,

- Autor opini o artykule,
- Autor akceptacji opinii o artykule,
- Recenzent,
- Administrator.

## 2.4.2. Diagramy przypadków użycia

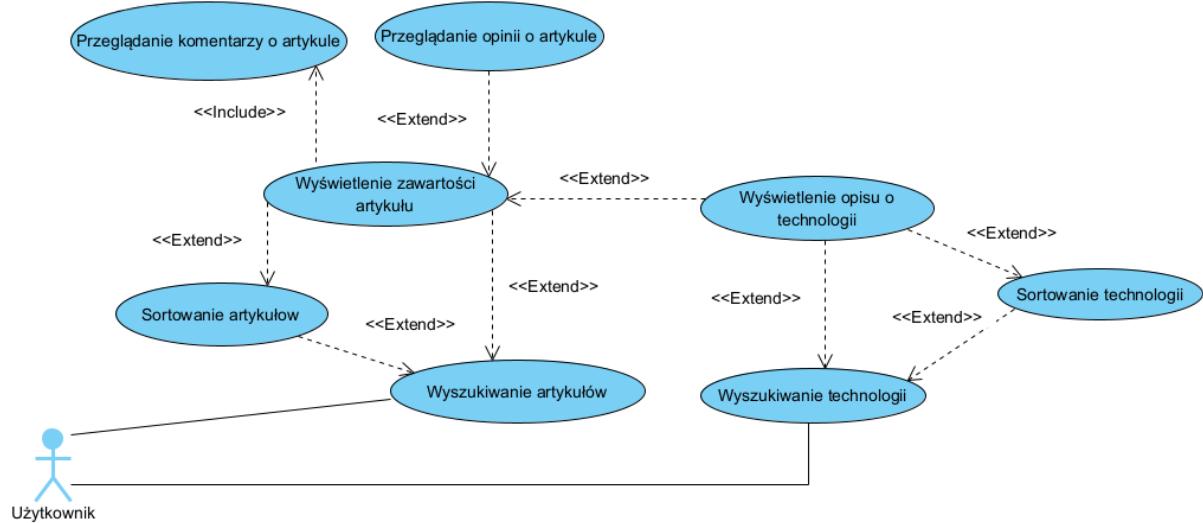
Na podstawie zdefiniowanych wcześniej wymagań funkcjonalnych oraz zidentyfikowanych aktorów, sporządzono diagramy przypadków użycia. Zdecydowano się na zastosowanie dziedziczenia między aktorami, gdyż opracowane role charakteryzują się tym, że posiadają wiele wspólnych uprawnień (przypadków użycia) i działanie to pozwala na zmniejszenie złożoności systemu. Dziedziczenie aktorów zostało przedstawione na rysunku 2.3.



Rys. 2.3: Dziedziczenie między aktorami

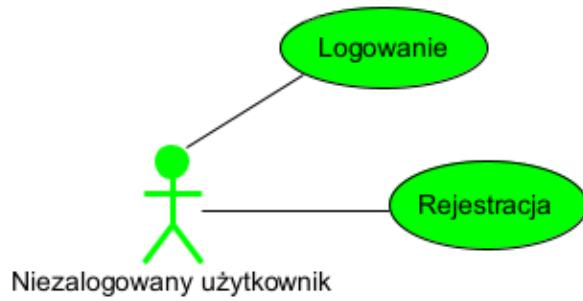
Źródło: opracowanie własne

Z powody dosyć dużej złożoności systemu pod względem liczby wymagań funkcjonalnych i liczby ról, dla zachowania czytelności, stworzono osobne diagramy dla każdego aktora. Poniżej znajdują się diagramy przypadków użycia dla poszczególnych aktorów.



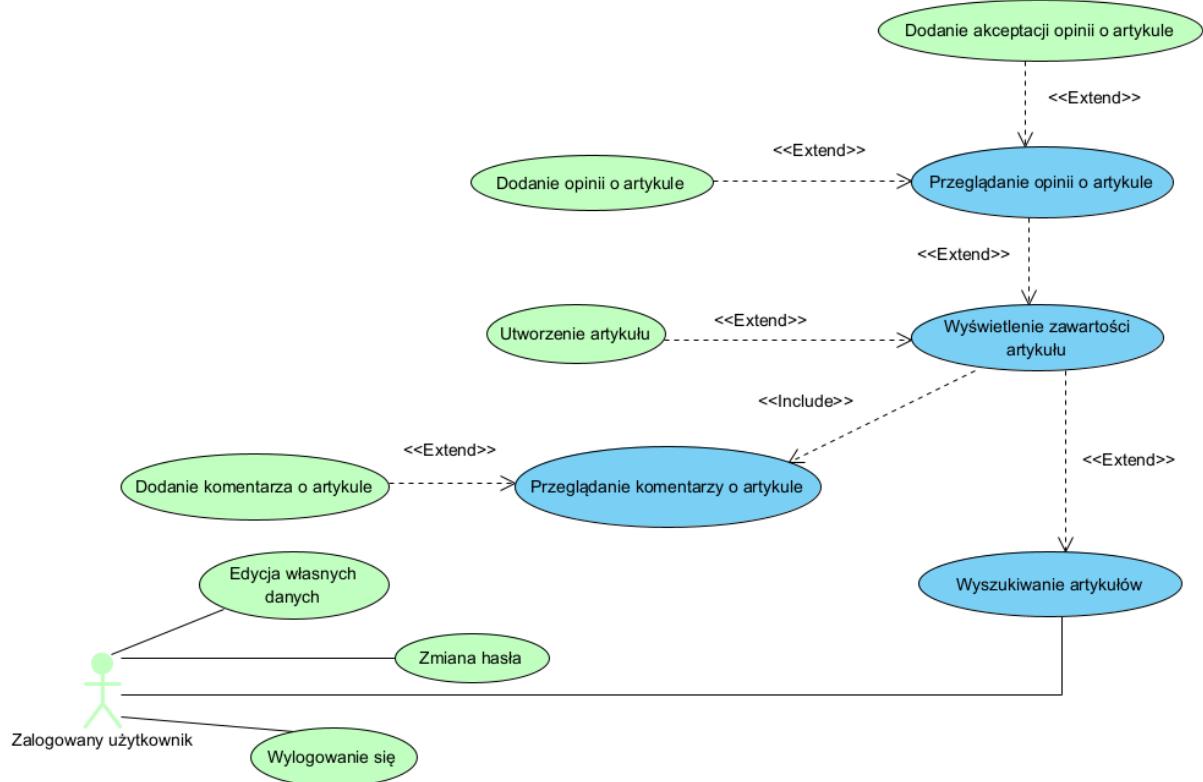
Rys. 2.4: Diagram przypadków użycia dla każdego użytkownika

Źródło: opracowanie własne



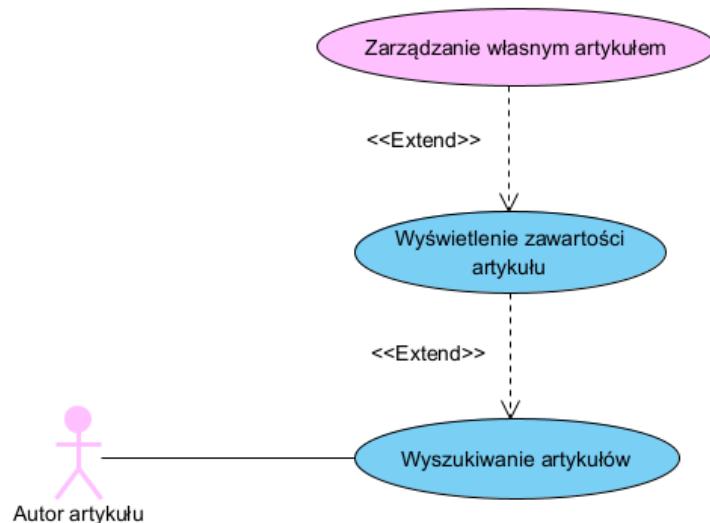
Rys. 2.5: Diagram przypadków użycia dla niezalogowanego użytkownika

Źródło: opracowanie własne



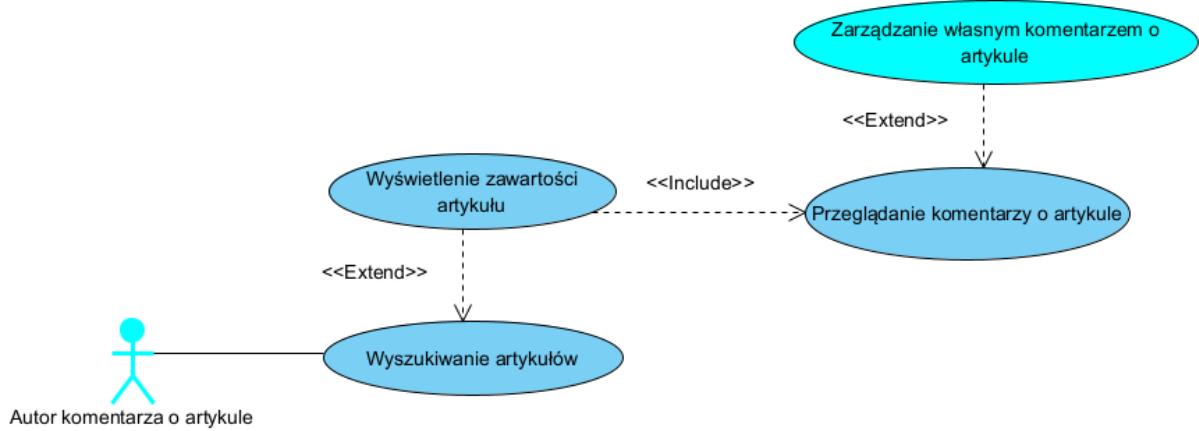
Rys. 2.6: Diagram przypadków użycia dla zalogowanego użytkownika

Źródło: opracowanie własne



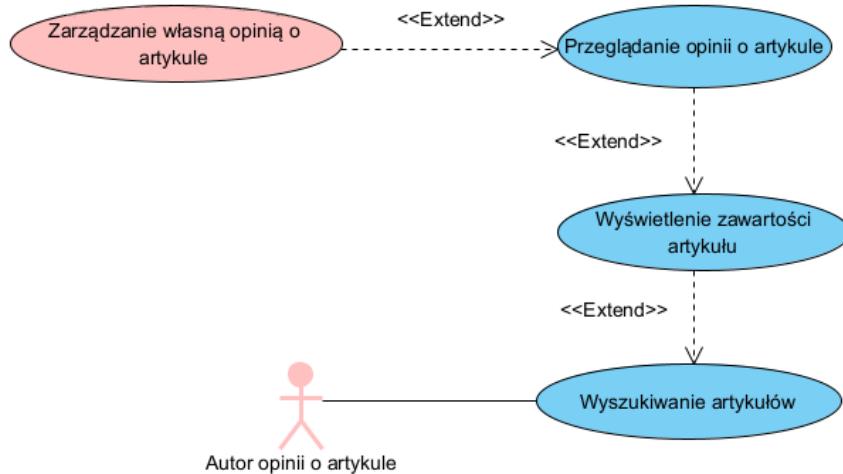
Rys. 2.7: Diagram przypadków użycia dla autora artykułu

Źródło: opracowanie własne



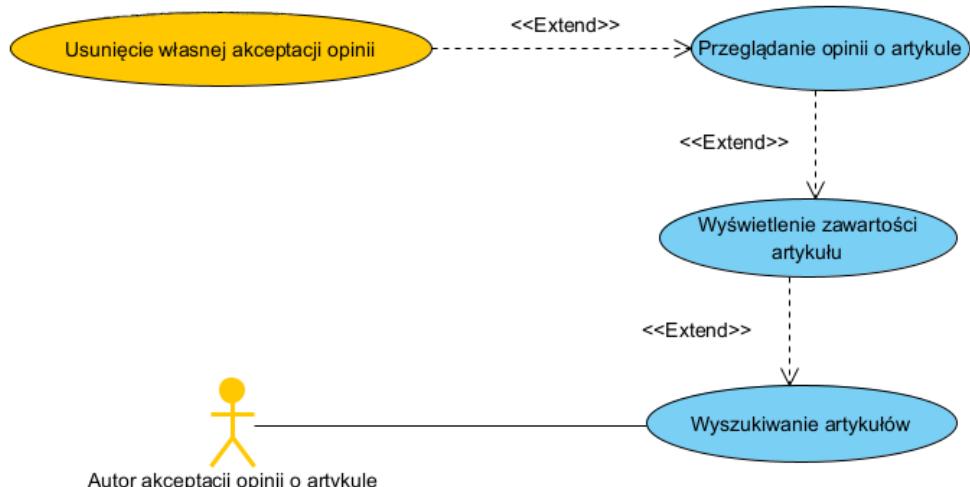
Rys. 2.8: Diagram przypadków użycia dla autora komentarza o artykule

Źródło: opracowanie własne



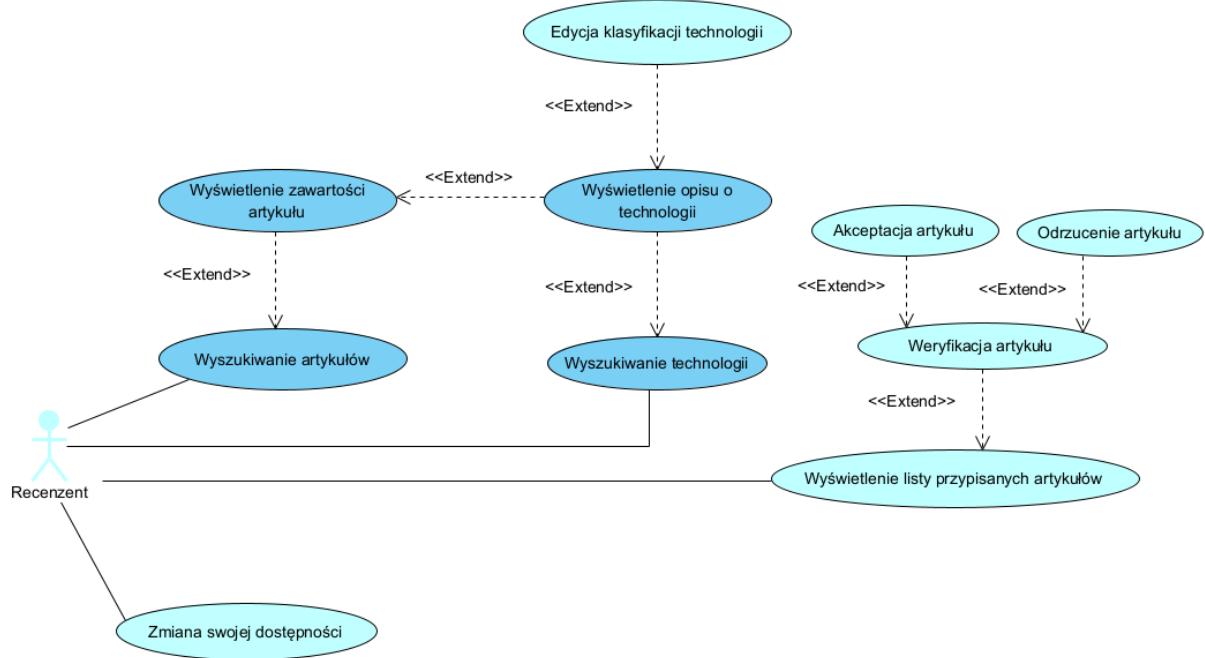
Rys. 2.9: Diagram przypadków użycia dla autora opinii o artykule

Źródło: opracowanie własne



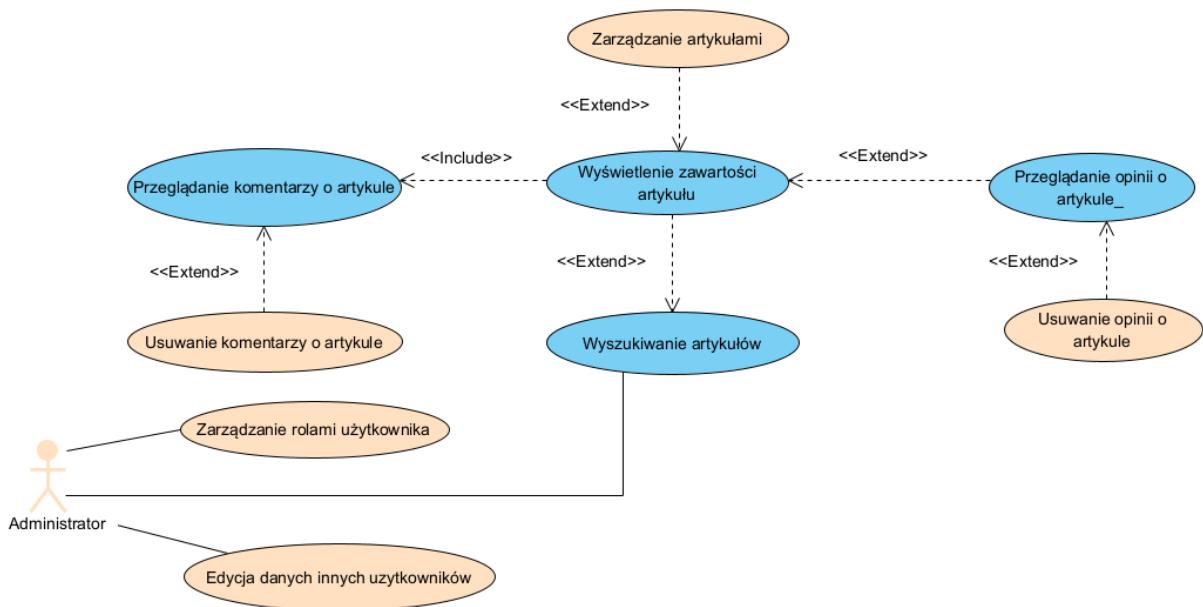
Rys. 2.10: Diagram przypadków użycia dla autora akceptacji opinii artykule

Źródło: opracowanie własne



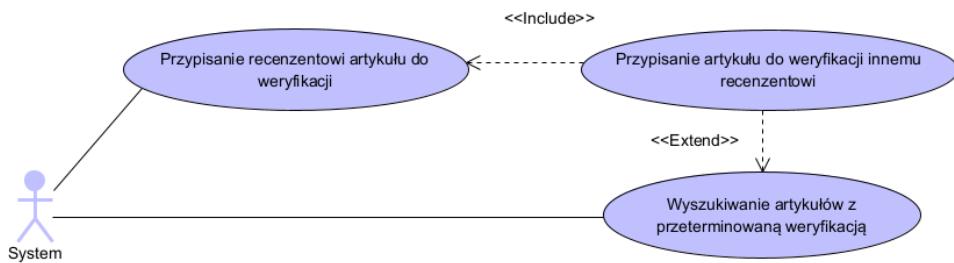
Rys. 2.11: Diagram przypadków użycia dla recenzenta

Źródło: opracowanie własne



Rys. 2.12: Diagram przypadków użycia dla administratora

Źródło: opracowanie własne



Rys. 2.13: Diagram przypadków użycia dla systemu

Źródło: opracowanie własne

### 2.4.3. Opisy przypadków użycia

Dla każdego przypadku użycia zdefiniowanego w poprzednim punkcie, zrobiono szczegółowe opisy w postaci tabel, które składają się z następujących informacji:

- Nazwa,
- Cel,
- Występujący aktorzy,
- Warunki początkowe,
- Warunki końcowe,
- Przebieg,
- Alternatywne przebiegi.

Dla każdego przypadku użycia zrobiono jeszcze prototypy ekranu użytkownika, aby zaplanować wygląd interfejsu użytkownika oraz aby łatwiej można było pojąć opisy przypadków użycia.

Tab. 2.1: Opis przypadku użycia - rejestracja

Nazwa:	Rejestracja
Cel:	Utworzenie konta zwykłego użytkownika
Aktorzy:	Niezalogowany użytkownik
Warunki początkowe:	Użytkownik jest niezalogowany
Warunki końcowe:	Założenie konta zwykłego użytkownika
Przebieg:	<ol style="list-style-type: none"> <li>Użytkownik kliką przycisk „Rejestracja” na nagłówku strony (Rys. 2.14),</li> <li>Aplikacja wyświetla formularz do rejestracji (Rys. 2.15),</li> <li>Użytkownik wypełnia pola w formularzu,</li> <li>Użytkownik kliką przycisk „Zarejestruj”,</li> <li>Serwis przeprowadza validację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>Po pozytywnej validacji danych, aplikacja wyświetla regulamin serwisu,</li> <li>Użytkownik przegląda cały regulamin i go akceptuje poprzez kliknięcie w pole wyboru „Akceptuję regulamin” (Rys. 2.16),</li> <li>Serwis tworzy nowo konto.</li> </ol>

Alternatywny przebieg:	<p>Użytkownik anuluje rejestrację</p> <p>3a. Użytkownik kliką przycisk „X” albo kliką w obszar poza formularzem,</p> <p>4a. Wyłączenie formularza.</p>
Alternatywny przebieg:	<p>Wprowadzone przez użytkownika dane nie przeszły walidacji</p> <p>6b. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</p> <p>7b. Powrót do punktu 3.</p>
Alternatywny przebieg:	<p>Użytkownik nie akceptuje regulaminu</p> <p>7c. Użytkownik nie akceptuje regulaminu poprzez kliknięcie w pole wyboru „Nie akceptuje regulaminu”, kliką przycisk „X” albo kliką w obszar poza formularzem,</p> <p>8c. Wyłączenie formularza.</p>

Logowanie

Rejestracja

Artykuły

Technologie

Rys. 2.14: Nagłówek strony niezalogowanego użytkownika

Źródło: opracowanie własne

Formularz rejestracyjny z polem do dodania avatara.

Formularz rejestracyjny (Rejestracja) składa się z następujących elementów:

- Pole imię (obowiązkowe, oznaczone gwiazdką \*)
- Pole nazwisko (obowiązkowe, oznaczone gwiazdką \*)
- Pole nazwa użytkownika (obowiązkowe, oznaczone gwiazdką \*)
- Pole pseudonim (obowiązkowe, oznaczone gwiazdką \*)
- Pole e-mail (obowiązkowe, oznaczone gwiazdką \*)
- Pole hasło (obowiązkowe, oznaczone gwiazdką \*)
- Pole powtórz hasło (obowiązkowe, oznaczone gwiazdką \*)
- Pole do dodania avatara (zapisane jako "K")
- Pole "Dodaj avatar" z przyciskiem "+"
- Konferencyjne przycisk "Zarejestruj"

\* - pole wymagane

Rys. 2.15: Formularz rejestracji

Źródło: opracowanie własne

X

# Regulamin

**1. Ogólne zasady**

Użytkownicy nie powinni umieszczać w serwisie treści zawierających wulgaryzmy, będących spamem lub zawierających linki do niebezpiecznych stron (wirusy itp.). W pierwszym przypadku treści te będą usuwane przez administratorów, a w przypadku skrajnych lub powtarzających się takich działań, konto użytkownika odpowiadającego za tego typu treści może zostać zablokowane na jakiś czas (zwykle 1 dzień) lub na stałe.

**2. Artykuły**

Użytkownicy nie powinni publikować w artykułach informacji nieprawdziwych, nieaktualnych, tekst nie powinien zawierać błędów, zamieszczane rysunki powinny być czytelne, a kod powinien być wstawiony za pomocą pól specjalnie do tego przeznaczonych.  
W przypadku niezastosowania się użytkownika do powyższych reguł, artykuł zostanie zedytorowany przez administratora w przypadku wymogu drobnych poprawek, a w przypadku dużych zmian, artykuł taki zostanie wycofany z serwisu, autor artykułu będzie musiał poprawić błędy i zmieniony artykuł przejdzie jeszcze raz weryfikację przez automatycznie wybranego recenzenta.

Akceptuję regulamin  
 Nie akceptuję regulaminu

Rys. 2.16: Formularz służący do akceptacji regulaminu

Źródło: opracowanie własne

Tab. 2.2: Opis przypadku użycia - logowanie

Nazwa:	Logowanie
Cel:	Zalogowanie się użytkownika
Aktorzy:	Niezalogowany użytkownik
Warunki początkowe:	Użytkownik jest niezalogowany
Warunki końcowe:	Zalogowanie się użytkownika

Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Logowanie” na nagłówku strony (Rys. 2.14),</li> <li>2. Aplikacja wyświetla formularz do logowania (Rys. 2.17),</li> <li>3. Użytkownik wypełnia pola w formularzu,</li> <li>4. Użytkownik kliką przycisk „Zaloguj”,</li> <li>5. Serwis przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>6. Po pozytywnej walidacji danych, użytkownik zostaje zalogowany do systemu i będzie mógł korzystać z funkcji systemu w zakresie adekwatnym do jego uprawnień. Zmienia się wygląd nagłówka strony (Rys. 2.18),</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje logowanie</p> <ol style="list-style-type: none"> <li>3a. Użytkownik kliką przycisk „X” lub kliką w obszar poza formularzem,</li> <li>4a. Wyłączenie formularza.</li> </ol>
Alternatywny przebieg:	<p>Wprowadzone przez użytkownika dane nie przeszły walidacji</p> <ol style="list-style-type: none"> <li>6b. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</li> <li>7b. Powrót do punktu 3.</li> </ol>
Alternatywny przebieg:	<p>Hasło użytkownika zostało wcześniej zresetowane (puste hasło)</p> <ol style="list-style-type: none"> <li>7c. Aplikacja wyświetla formularz do ustawienia hasła (Rys. 2.19)</li> <li>8c. Użytkownik podaje hasło,</li> <li>9c. Użytkownik kliką przycisk „Zapisz”,</li> <li>10c. Serwis sprawdza, czy podano hasło oraz jeśli podano, to czy jest ono zgodne z narzuconym rozmiarem i formatem,</li> <li>11c. Po pozytywnej walidacji hasła, hasło użytkownika zostaje zmienione.</li> </ol>
Alternatywny przebieg:	<p>Wprowadzone hasło po resecie hasła nie przeszło walidacji</p> <ol style="list-style-type: none"> <li>11ca. Aplikacja zmienia kolor obramowania pola z haślem, na czerwono oraz wyświetla pod tym polem adekwatny komunikat o błędzie</li> <li>12ca. Powrót do punktu 8c,</li> </ol>

Formularz logowania zatytułowany "Logowanie". W środku znajdują się dwa pola do wpisania danych: "Nazwa użytkownika" i "Hasło", a także niebieska przycisk "Zaloguj" na dole.

Rys. 2.17: Formularz logowania

Źródło: opracowanie własne

Rys. 2.18: Nagłówek strony zalogowanego użytkownika

Źródło: opracowanie własne

Formularz do ustawiania hasła zatytułowany "Ustawianie hasła". W środku znajdują się pole do wpisania "Hasło" i niebieska przycisk "Zapisz" na dole.

Rys. 2.19: Formularz do ustawienia hasła po zresetowaniu hasła

Źródło: opracowanie własne

Tab. 2.3: Opis przypadku użycia - wylogowanie

Nazwa:	Wylogowanie
Cel:	Wylogowanie się użytkownika
Aktorzy:	Zalogowany użytkownik
Warunki początkowe:	Użytkownik jest zalogowany
Warunki końcowe:	Wylogowanie użytkownika z serwisu

Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką na swój awatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta (Rys. 2.20),</li> <li>2. Użytkownik kliką przycisk „Wyloguj się”,</li> <li>3. Aplikacja wyświetla komunikat o wylogowaniu się,</li> <li>4. Następuje wylogowanie użytkownika oraz zmienia się wygląd nagłówka strony (Rys. 2.14).</li> </ol>
-----------	---



Rys. 2.20: Nagłówek strony zalogowanego użytkownika z otwartymi opcjami dotyczącymi konta

Źródło: opracowanie własne

Tab. 2.4: Opis przypadku użycia - edycja własnych danych

Nazwa:	Edycja własnych danych
Cel:	Zapisanie danych użytkownika zgodnie z wprowadzonymi przez niego zmianami
Aktorzy:	Zalogowany użytkownik
Warunki początkowe:	Użytkownik jest zalogowany
Warunki końcowe:	Zapisanie zmienionych danych użytkownika
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką na swój awatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta (Rys. 2.20),</li> <li>2. Użytkownik kliką przycisk „Dane użytkownika”,</li> <li>3. Aplikacja wyświetla panel z danymi użytkownika (Rys 2.21),</li> <li>4. Użytkownik kliką przycisk „Edytuj dane”,</li> <li>5. Serwis wyświetla formularz edycji danych (Rys 2.22),</li> <li>6. Użytkownik zmienia dane w formularzu,</li> <li>7. Użytkownik kliką przycisk „Zapisz”,</li> <li>8. Serwis przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>9. Po pozytywnej walidacji danych, aplikacja zapisuje zmienione dane użytkownika.</li> </ol>

Alternatywny przebieg:	<p>Użytkownik anuluje edycję danych</p> <p>6a. Użytkownik kliką przycisk „Anuluj”,      7a. Aplikacja wyświetla panel z danymi użytkownika (Rys 2.21).</p>
Alternatywny przebieg:	<p>Wprowadzone przez użytkownika dane nie przeszły walidacji</p> <p>9b. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,      10b. Powrót do punktu 6.</p>

**IT Tech** Technologie IT

kamil\_dywan

Artykuły

Technologie

### Dane użytkownika

Imię	Kamil	
Nazwisko	Dywan	
Nazwa użytkownika	kamil.dywan	Avatar 
Pseudonim	kamil_dywan	
E-mail	kamil.dywan@mail.pl	

[Edytuj dane](#)

Rys. 2.21: Panel z danymi użytkownika

Źródło: opracowanie własne

The screenshot shows a user profile editing interface. At the top, there's a header with 'IT Tech Technologie IT' and a user icon labeled 'kamil\_dywan'. Below the header, there are two tabs: 'Artykuły' and 'Technologie'. The main area is titled 'Dane użytkownika' (User Data). It contains fields for 'Imię' (Name), 'Nazwisko' (Surname), 'Nazwa użytkownika' (Username), 'Pseudonim' (Nickname), and 'E-mail' (Email). Each field has a corresponding input box. To the right of the username and nickname boxes is an 'Avatar' section featuring a blue square placeholder with a white letter 'K' and a file selection box labeled 'avatar.png +'. At the bottom of the form are two buttons: 'Zapisz' (Save) in blue and 'Anuluj' (Cancel) in grey.

Rys. 2.22: Panel umożliwiający użytkownikowi edycję własnych danych

Źródło: opracowanie własne

Tab. 2.5: Opis przypadku użycia - zmiana hasła

Nazwa:	Zmiana hasła
Cel:	Zmiana hasła zgodnie z wprowadzoną przez użytkownika wartością
Aktorzy:	Zalogowany użytkownik
Warunki początkowe:	Użytkownik jest zalogowany
Warunki końcowe:	Zapisanie zmienionego hasła użytkownika
Przebieg:	<ol style="list-style-type: none"> <li>Użytkownik klikna na swój avatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta (Rys. 2.20),</li> <li>Użytkownik klikna przycisk „Zmień hasło”,</li> <li>Aplikacja wyświetla formularz do zmiany hasła (Rys. 2.23)</li> <li>Użytkownik podaje aktualne oraz nowe hasło,</li> <li>Użytkownik klikna przycisk „Zapisz”,</li> <li>Serwis sprawdza, czy podano hasła, czy podane aktualne hasło jest prawidłowe oraz czy podane nowe hasło spełnia wymagania pod względem narzuconego rozmiaru i formatu danych,</li> <li>Po pozytywnej walidacji hasło użytkownika zostaje zmienione.</li> </ol>
Alternatywny przebieg:	Użytkownik anuluje zmianę hasła 4a. Użytkownik klikna przycisk „Anuluj”, 5a. Aplikacja wyłącza formularz do zmiany hasła.

Alternatywny przebieg:	<p>Wprowadzone przez użytkownika dane nie przeszły walidacji</p> <p>7b. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</p> <p>8b. Powrót do punktu 4.</p>
------------------------	---

Formularz zmiany hasła z dwoma polami tekstowymi: 'Aktualne hasło' i 'Nowe hasło', oraz dwiema przyciskami: 'Zapisz' i 'Anuluj'.

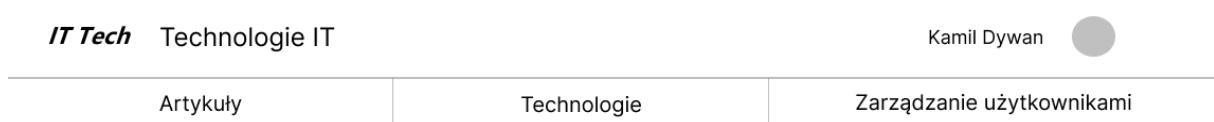
Rys. 2.23: Formularz zmiany hasła

Źródło: opracowanie własne

Tab. 2.6: Opis przypadku użycia - edycja danych innych użytkowników

Nazwa:	Edycja danych innych użytkowników
Opis:	Zapisanie danych innego użytkownika zgodnie z wprowadzonymi zmianami
Aktorzy:	Administrator
Warunki początkowe:	Użytkownik posiadający rolę administratora jest zalogowany
Warunki końcowe:	Zapisanie zmienionych danych innego użytkownika

Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Zarządzanie użytkownikami” znajdujący się w menu głównym strony (Rys. 2.24),</li> <li>2. Aplikacja wyświetla podstronę z wyszukiwarką użytkowników,</li> <li>3. Użytkownik wpisuje nazwę użytkownika (np. kamil),</li> <li>4. Użytkownik kliką przycisk „Szukaj”,</li> <li>5. Serwis wyszukuje użytkowników,</li> <li>6. Aplikacja wyświetla listę wyszukanych użytkowników (Rys 2.25)</li> <li>7. Użytkownik kliką na jednego z innych użytkowników,</li> <li>8. Aplikacja wyświetla panel z danymi wybranego użytkownika (Rys 2.26),</li> <li>9. Użytkownik kliką przycisk „Edytuj dane”,</li> <li>10. Serwis wyświetla formularz edycji danych (Rys 2.27),</li> <li>11. Użytkownik zmienia dane w formularzu,</li> <li>12. Użytkownik kliką przycisk „Zapisz”,</li> <li>13. Serwis przeprowadza validację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>14. Po pozytywnej walidacji danych, aplikacja zapisuje zmienione dane innego użytkownika.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik resetuje hasło innego użytkownika</p> <ol style="list-style-type: none"> <li>9a. Użytkownik kliką przycisk „Reset hasła”,</li> <li>10a. Aplikacja zastępuje hasło wybranego użytkownika pustym ciągiem znaków,</li> <li>11a. Powrót do punktu 8.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje edycję danych</p> <ol style="list-style-type: none"> <li>11b. Użytkownik kliką przycisk „Anuluj”,</li> <li>12b. Aplikacja wyświetla panel z danymi użytkownika (Rys. 2.26).</li> </ol>
Alternatywny przebieg:	<p>Wprowadzone przez użytkownika dane nie przeszły walidacji</p> <ol style="list-style-type: none"> <li>14c. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</li> <li>15c. Powrót do punktu 11.</li> </ol>



Rys. 2.24: Nagłówek strony administratora

Źródło: opracowanie własne

The screenshot shows a search interface titled "Wyszukiwanie użytkowników". At the top, there is a search input field containing "kamil" and a blue "Szukaj" button. Below the input field is a table with three columns: "Nazwa użytkownika", "Pseudonim", and "Imię i nazwisko". The table contains four rows of data:

Nazwa użytkownika	Pseudonim	Imię i nazwisko
kamil.dywan	kamil_dywan	Kamil Dywan
kamil.nowak	kamil_nowak	Kamil Nowak
kamil.kowalski	kamil_kowalski	Kamil Kowalski

At the bottom of the table are navigation buttons: '<', '1', '2', '...', '8', and '>'.

Rys. 2.25: Panel do wyszukiwania użytkowników

Źródło: opracowanie własne



## Dane użytkownika

Imię	Kamil	
Nazwisko	Dywan	
Nazwa użytkownika	kamil.dywan	Avatar
Pseudonim	kamil_dywan	
E-mail	kamil.dywan@mail.com	

[Reset hasła](#)[Edytuj dane](#)

## Role użytkownika

Recenzent	<input checked="" type="checkbox"/>
Administrator	<input type="checkbox"/>

Rys. 2.26: Panel z danymi użytkownika widziany przez administratora

Źródło: opracowanie własne

Dane użytkownika

Imię	Kamil
Nazwisko	Dywan
Nazwa użytkownika	kamil.dywan
Pseudonim	kamil_dywan
E-mail	kamil.dywan@mail.com

Avatar

**K**

avatar.png +

Reset hasła

Zapisz

Anuluj

## Role użytkownika

Recenzent	<input checked="" type="checkbox"/>
Administrator	<input type="checkbox"/>

Rys. 2.27: Panel umożliwiający administratorowi edycję danych użytkownika

Źródło: opracowanie własne

Tab. 2.7: Opis przypadku użycia - zarządzanie rolami użytkownika

Nazwa:	Zarządzanie rolami innego użytkownika
Cel:	Zmiana przypisanych ról innego użytkownika
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlony panel z danymi innego użytkownika (Rys 2.26)
Warunki końcowe:	Zmiana przypisanych ról innego użytkownika

Przebieg:	<p>Inny użytkownik posiada wybraną rolę (np. Recenzent na Rys 2.26)</p> <ol style="list-style-type: none"> <li>1. Użytkownik kliką na pole wyboru przy wybranej roli (jeśli przy roli jest zaznaczone pole wyboru, to oznacza to, że inny użytkownik ma przypisaną taką rolę, a jeśli pole wyboru będzie puste, to oznacza, że inny użytkownik nie posiada takiej roli),</li> <li>2. Pole wyboru staje się puste,</li> <li>3. Aplikacja usuwa przypisanie roli innego użytkownika.</li> </ol>
Alternatywny przebieg:	<p>Inny użytkownik nie posiada wybranej roli (np. Administrator na Rys 2.26)</p> <ol style="list-style-type: none"> <li>2a. Pole wyboru zostaje zaznaczone</li> <li>3a. Aplikacja przypisuje wybraną rolę innemu użytkownikowi.</li> </ol>

Tab. 2.8: Opis przypadku użycia - zmiana własnej dostępności

Nazwa:	Zmiana własnej dostępności
Cel:	Zmiana własnej dostępności w przypadku przypisywania artykułów do weryfikacji
Aktorzy:	Recenzent
Warunki początkowe:	Użytkownik posiadający rolę recenzenta jest zalogowany
Warunki końcowe:	Zmiana dostępności użytkownika

Przebieg:	<p>Użytkownik ma własną dostępność ustawioną na „Dostępny”</p> <ol style="list-style-type: none"> <li>1. Użytkownik klikna na swój awatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta,</li> <li>2. Użytkownik klikna przycisk „Dostępny” (Rys. 2.28),</li> <li>3. Aplikacja wyświetla formularz do ustawienia daty, do której użytkownik będzie miał ustawioną niedostępność,</li> <li>4. Użytkownik klikna na pole do wypełnienia daty,</li> <li>5. Serwis wyświetla formularz do wybrania daty (Rys. 2.29),</li> <li>6. Użytkownik wybiera datę (np. 18.08.2014),</li> <li>7. Aplikacja wyświetla ustawioną datę,</li> <li>8. Użytkownik klikna przycisk „Zapisz”,</li> <li>9. Serwis sprawdza, czy użytkownik podał datę,</li> <li>10. Użytkownik podał datę, zatem aplikacja zmienia dostępność recenzenta na „Niedostępny” (Rys. 2.30), co będzie skutkowało tym, że takiemu recenzentowi nie będą przypisywane artykuły do weryfikacji. Po upłynięciu daty podanej przez użytkownika, status obecności tego użytkownika zostanie automatycznie zmieniony na „Dostępny”.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje zmianę dostępności</p> <ol style="list-style-type: none"> <li>4a. Użytkownik klikna w „X” w prawym górnym rogu lub kilka w obszar poza formularzem,</li> <li>5a. Aplikacja wyłącza formularz do zmiany dostępności.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie podał daty</p> <ol style="list-style-type: none"> <li>10b. Aplikacja zmienia kolor obramowania pola z podaniem daty na czerwono oraz wyświetla pod tym polem komunikat „Pole wymagane”,</li> <li>11b. Powrót do punktu 4.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik ma ustawioną własną dostępność na „Niedostępny”</p> <ol style="list-style-type: none"> <li>2c. Użytkownik klikna przycisk „Niedostępny” (Rys. 2.30),</li> <li>3c. Aplikacja zmienia dostępność recenzenta na „Dostępny” (Rys. 2.28), czyli takiemu recenzentowi będą teraz przypisywane artykuły do weryfikacji.</li> </ol>



Rys. 2.28: Nagłówek strony dostępnego recenzenta z otwartymi opcjami dotyczącymi konta

Źródło: opracowanie własne

The form is titled 'Ustawienie niedostępności' (Setting Unavailability). It includes a date input field labeled 'Do dnia' (Until) with the value '18/08/2014'. Below it is a calendar for August 2014, showing the 18th as the selected date. At the bottom of the form is a blue 'Zapisz' (Save) button.

Rys. 2.29: Formularz z ustawieniem daty, do której ma być ustwiona nieobecność recenzenta razem z otwartym oknem wyboru daty

Źródło: opracowanie własne



Rys. 2.30: Nagłówek strony niedostępnego recenzenta z otwartymi opcjami dotyczącymi konta

Źródło: opracowanie własne

Tab. 2.9: Opis przypadku użycia - wyszukiwanie artykułów

Nazwa:	Wyszukiwanie artykułów
Cel:	Wyszukiwanie artykułów według podanych przez użytkownika kryteriów i wyświetlenie wyników tego wyszukiwania w postaci listy artykułów
Aktorzy:	Użytkownik
Warunki początkowe:	Brak
Warunki końcowe:	Wyświetlenie listy wyszukanych artykułów
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Artykuły” znajdujący się w panelu nawigacji strony (Rys. 2.14),</li> <li>2. Aplikacja wyświetla widok związany z wyszukiwaniem artykułów,</li> <li>3. Użytkownik uzupełnia kryteria wyszukiwania,</li> <li>4. Użytkownik kliką przycisk „Szukaj”,</li> <li>5. Serwis wyszukuje artykuły,</li> <li>6. Aplikacja wyświetla listę wyszukanych artykułów (Rys. 2.31), przy czym własne artykuły są oznaczone niebieskim prostokątem.</li> </ol>

Artykuły	Technologie																												
<b>Wyszukiwanie artykułów</b> <span style="float: right;">Dodaj artykuł</span>																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Tytuł</td><td colspan="2">Java</td></tr> <tr><td>Autor</td><td colspan="2">kamil.dywan</td></tr> <tr><td>Kategoria technologii</td><td colspan="2">Język programowania ▾</td></tr> <tr><td>Technologia</td><td colspan="2">Java ▾</td></tr> <tr><td>Dostawca technologii</td><td colspan="2">Oracle ▾</td></tr> <tr><td>Data powstania</td><td>01.01.1995</td><td>01.07.2000</td></tr> <tr><td>Data modyfikacji</td><td>01.01.2022</td><td>01.12.2022</td></tr> </table>		Tytuł	Java		Autor	kamil.dywan		Kategoria technologii	Język programowania ▾		Technologia	Java ▾		Dostawca technologii	Oracle ▾		Data powstania	01.01.1995	01.07.2000	Data modyfikacji	01.01.2022	01.12.2022							
Tytuł	Java																												
Autor	kamil.dywan																												
Kategoria technologii	Język programowania ▾																												
Technologia	Java ▾																												
Dostawca technologii	Oracle ▾																												
Data powstania	01.01.1995	01.07.2000																											
Data modyfikacji	01.01.2022	01.12.2022																											
<input type="button" value="Szukaj"/>																													
Sortowanie <span style="border: 1px solid #ccc; padding: 2px 10px; width: 100px; display: inline-block;">▼</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">&lt;</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">1</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">2</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">...</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">8</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">&gt;</span>																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Java - poradnik</td><td>Java</td><td>Język progr.</td><td><span style="color: green;">●</span></td><td>Adam Nowak</td><td>21:00, 01.01.2022</td><td><span style="color: yellow;">★</span> 50%</td></tr> <tr><td>Podstawy Javy</td><td>Java</td><td>Język progr.</td><td><span style="color: grey;">●</span></td><td>Kamil Dywan</td><td>20:30, 04.03.2022</td><td><span style="color: yellow;">★</span> 62%</td></tr> <tr><td>Wersje Javy</td><td>Java</td><td>Język progr.</td><td><span style="color: grey;">●</span></td><td>Kamil Dywan</td><td>22:00, 02.05.2022</td><td><span style="color: yellow;">★</span> 80%</td></tr> <tr><td>Java - strumienie</td><td>Java</td><td>Język progr.</td><td><span style="color: lime;">●</span></td><td>Michał Nowak</td><td>19:30, 06.07.2022</td><td><span style="color: yellow;">★</span> 98%</td></tr> </table>		Java - poradnik	Java	Język progr.	<span style="color: green;">●</span>	Adam Nowak	21:00, 01.01.2022	<span style="color: yellow;">★</span> 50%	Podstawy Javy	Java	Język progr.	<span style="color: grey;">●</span>	Kamil Dywan	20:30, 04.03.2022	<span style="color: yellow;">★</span> 62%	Wersje Javy	Java	Język progr.	<span style="color: grey;">●</span>	Kamil Dywan	22:00, 02.05.2022	<span style="color: yellow;">★</span> 80%	Java - strumienie	Java	Język progr.	<span style="color: lime;">●</span>	Michał Nowak	19:30, 06.07.2022	<span style="color: yellow;">★</span> 98%
Java - poradnik	Java	Język progr.	<span style="color: green;">●</span>	Adam Nowak	21:00, 01.01.2022	<span style="color: yellow;">★</span> 50%																							
Podstawy Javy	Java	Język progr.	<span style="color: grey;">●</span>	Kamil Dywan	20:30, 04.03.2022	<span style="color: yellow;">★</span> 62%																							
Wersje Javy	Java	Język progr.	<span style="color: grey;">●</span>	Kamil Dywan	22:00, 02.05.2022	<span style="color: yellow;">★</span> 80%																							
Java - strumienie	Java	Język progr.	<span style="color: lime;">●</span>	Michał Nowak	19:30, 06.07.2022	<span style="color: yellow;">★</span> 98%																							
Sortowanie <span style="border: 1px solid #ccc; padding: 2px 10px; width: 100px; display: inline-block;">▼</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">&lt;</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">1</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">2</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">...</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">8</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">&gt;</span>																													

Rys. 2.31: Panel do wyszukiwania artykułów z podanymi kryteriami wyszukiwania oraz listą wyszukanych artykułów

Źródło: opracowanie własne

Tab. 2.10: Opis przypadku użycia - sortowanie artykułów

Nazwa:	Sortowanie artykułów
Cel:	Posortowanie wyszukanych artykułów według opcji sortowania wybranej przez użytkownika oraz wyświetlenie listy tych posortowanych artykułów
Aktorzy:	Użytkownik
Warunki początkowe:	Wyświetlona lista wyszukanych artykułów (Rys. 2.31)
Warunki końcowe:	Wyświetlenie listy posortowanych artykułów
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką na rozsuwaną listę „Sortowanie”,</li> <li>2. Użytkownik wybiera jedną z opcji sortowania (Rys. 2.32),</li> <li>3. Aplikacja sortuje artykuły według wybranej opcji sortowania,</li> <li>4. Aplikacja wyświetla listę posortowanych artykułów, przy czym własne artykuły są oznaczone niebieskim prostokątem.</li> </ol>

Sortowanie ▼
Nazwa: malejąco
Nazwa: rosnąco
Technologia: malejąco
Technologia: rosnąco
Autor: malejąco
Autor: rosnąco
Data powstania: malejąco
Data powstania: rosnąco
Data modyfikacji: malejąco
Data modyfikacji: rosnąco
Średnia ocena: malejąco
Średnia ocena: rosnąco
Liczba opinii: malejąco
Liczba opinii: rosnąco

Rys. 2.32: Dostępne opcje sortowania artykułów

Źródło: opracowanie własne

Tab. 2.11: Opis przypadku użycia - wyświetlenie zawartości artykułu

Nazwa:	Wyświetlenie zawartości artykułu
Cel:	Wyświetlenie zawartości artykułu
Aktorzy:	Użytkownik
Warunki początkowe:	Wyświetlona lista wyszukanych artykułów (Rys. 2.31)
Warunki końcowe:	Wyświetlenie zawartości artykułu
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik klikna na jeden z wyszukanych artykułów,</li> <li>2. Aplikacja wyświetla zawartość artykułu oraz komentarze dotyczące tego artykułu (Rys. 2.33).</li> </ol>

Artykuły	Technologie
<p>Kamil Dywan</p> <p>Data utworzenia: 14.07.2022, 00:32:45</p> <p>Data modyfikacji: 14.07.2022, 00:32:45</p> <p>Technologia → Język programowania → Java</p>  <p>Dostawca: Oracle</p> <p>Opis:</p> <p>Java jest to wspólnie dostępny, obiektowy język programowania. Język ten jest niezwykle popularny oraz uniwersalny i dlatego trudnym zadaniem byłoby znalezienie dziedziny na pograniczu informatyki, w której problemach nie można by było wykorzystać Javy. Jednym z powodów uniwersalności tego języka jest to, że programy napisane w Javie są uruchamiane na maszynie wirtualnej, co też powoduje, że aby uruchomić program napisany w Javie, wystarczy posiadać zaistalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.</p> <h3>Podstawy Javy</h3> <p>Java jest to wspólnie dostępny, oparty na klasach i mocno typowany obiektowy język programowania, który może być wykorzystywany w wielu dziedzinach. Został stworzony przez zespół pod kierownictwem Jamesa Goslinga z firmy Sun Microsystems, a pierwsze wydanie tego języka odbyło się w 1996 roku. Ważną cechą tego języka jest to, że przy komplikacji źródła kodu w nim napisanych, używana jest maszyna wirtualna nazywana JVM (ang. Java Virtual Machine).</p> <ol style="list-style-type: none"> <li>1. JVM</li> <li>Kod napisany w tym języku jest komplikowany do kodu bajtowego, który to następnie jest wykonywany przez maszynę wirtualną. Dodatkowo używana maszyna wirtualna sprawia, że język ten jest wieloplatformowy.</li> <li>...</li> <li>5. Zastosowania</li> </ol> <p>Z powodu na używaną maszynę wirtualną, programy napisane w Javie mogą być uruchomione na praktycznie wszystkich urządzeniach, co prowadzi do tego, że zastosowania tego języka mogą być bardzo rozległe. Przede wszystkim język jest wykorzystywany w aplikacjach Webowych jako backend oraz aplikacjach mobilnych. W przypadku zastosowań biznesowych język jest wykorzystywany przede wszystkim w rozwiązańach wysokopoziomowych przez korporacje i duże firmy. Język ten z powodu używanej maszyny wirtualnej jest wolniejszy niż np. C++, dlatego też jest najczęściej wykorzystywany w aplikacjach wysokopoziomowych.</p> <p>Platforma Mobilne Web Wysokopoziomowe rozwiązania Korporacyjne aplikacje</p> <p>6. Wersje</p> <p>1.0 - 1996 2.0 - grudzień 1998 - 1999 ... 9.0 - wrzesień 2017 11.0 - wrzesień 2018 15.0 - wrzesień 2020 17.0 - wrzesień 2021 7. Linki</p> <p>[1] <a href="https://docs.oracle.com/en/java/">https://docs.oracle.com/en/java/</a> - oficjalna dokumentacja</p> <p> 75%</p> <p><a href="#">Komentarze</a> <a href="#">Opinie</a> <a href="#">Opis technologii</a></p> <p><a href="#">Dodaj komentarz</a></p> <p> Kamil Dywan 15.07.2022, 00:32:45 </p> <p>Dobry artykuł <a href="#">Dodaj komentarz</a></p>	<p>Status: <b>Opublikowany</b></p> <p><a href="#">Edytuj</a> <a href="#">Usuń</a></p>

Rys. 2.33: Panel z zawartością artykułu

Źródło: opracowanie własne

Tab. 2.12: Opis przypadku użycia - utworzenie artykułu

Nazwa:	Utworzenie artykułu
Cel:	Utworzenie artykułu i wyświetlenie zawartości tego artykułu
Aktorzy:	Zalogowany użytkownik
Warunki początkowe:	Wyświetlony panel do wyszukiwania artykułów (Rys. 2.31)
Warunki końcowe:	Wyświetlenie zawartości utworzonego artykułu
Przebieg:	<p>Dodawanie artykułu</p> <ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Dodaj artykuł”,</li> <li>2. Aplikacja wyświetla formularz do dodania artykułu,</li> <li>3. Użytkownik wypełnia wszystkie pola w formularzu (Rys. 2.34)</li> <li>4. Użytkownik kliką przycisk „Zapisz”,</li> <li>5. Serwis przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>6. Po pozytywnej walidacji danych, aplikacja zapisuje stan artykułu,</li> <li>7. Użytkownik kliką przycisk „Wyślij do weryfikacji”,</li> <li>8. Serwis przesyła artykuł do weryfikacji (zapisuje artykuł w kolejce artykułów, do których mają być przydzielone recenzencji),</li> <li>9. Aplikacja wyświetla użytkownikowi wygląd utworzonego artykułu.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje dodanie artykułu</p> <ol style="list-style-type: none"> <li>3a. Użytkownik kliką przycisk „Anuluj”,</li> <li>4a. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie uzupełnia wszystkich pól formularza podczas dodawania artykułu</p> <ol style="list-style-type: none"> <li>6b. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</li> <li>7b. Przejście do punktu 3.</li> </ol>

Tytuł Kategoria  v

Java jest to współbieżny, oparty na klasach i mocno typowany obiektowy język programowania, który może być wykorzystywany w wielu dziedzinach. Został stworzony przez zespół pod kierownictwem Jamesa Goslinga z firmy Sun Microsystems, a pierwsze wydanie tego języka odbyło się w 1996 roku. Ważną cechą tego języka jest to, że przy komplikacji źródłek kodu w nim napisanych, używana jest maszyna wirtualna naazywana JVM (ang. Java Virtual Machine).

## 1. JVM

Kod napisany w tym języku jest kompilowany do kodu bajtowego, który to następnie jest wykonywany przez maszynę wirtualną. Dodatkowo używana maszyna wirtualna sprawia, że język ten jest wieloplatformowy.

...  
6. Wersje

1.0 - 1996  
 2.0 - grudzień 1998 - 1999  
 ...  
 9.0 - wrzesień 2017  
 11.0 - wrzesień 2018  
 15.0 - wrzesień 2020  
 17.0 - wrzesień 2021

## 7. Linki

[1] <https://docs.oracle.com/en/java/> - oficjalna dokumentacja

Rys. 2.34: Panel z formularzem dodania lub edytowania artykułu

Źródło: opracowanie własne

Tab. 2.13: Opis przypadku użycia - zarządzanie własnym artykułem

Nazwa:	Zarządzanie własnym artykułem
Cel:	Zmodyfikowanie lub usunięcie własnego artykułu
Aktorzy:	Autor artykułu
Warunki początkowe:	Wyświetlony panel do wyszukiwania artykułów (Rys. 2.31)
Warunki końcowe:	Zmodyfikowanie lub usunięcie artykułu

Przebieg:	<p>Edytowanie artykułu</p> <ol style="list-style-type: none"> <li>1. Użytkownik wyszukuje artykuły (Tab. 2.9),</li> <li>2. Użytkownik kliką na jeden z wyszukanych artykułów,</li> <li>3. Aplikacja wyświetla zawartość wybranego artykułu (Rys. 2.33),</li> <li>4. Użytkownik kliką przycisk „Edytuj”,</li> <li>5. Serwis wyświetla formularz do edycji artykułu (Rys. 2.34),</li> <li>6. Użytkownik modyfikuje pola w formularzu,</li> <li>7. Użytkownik kliką przycisk „Zapisz”,</li> <li>8. Aplikacja przeprowadza validację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>9. Po pozytywnej validacji danych, aplikacja zapisuje stan edytowanego artykułu,</li> <li>10. Użytkownik kliką przycisk „Wyślij do weryfikacji”,</li> <li>11. Serwis przesyła artykuł do weryfikacji (zapisuje artykuł w kolejce artykułów, do których mają być przydzieleni recenzenci),</li> <li>12. Aplikacja przedstawia użytkownikowi wygląd zmienionego artykułu.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje edytowanie artykułu</p> <ol style="list-style-type: none"> <li>6a. Użytkownik kliką przycisk „Anuluj”,</li> <li>7a. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie uzupełnia wszystkich pól formularza podczas edytowania artykułu</p> <ol style="list-style-type: none"> <li>9b. Aplikacja zmienia kolor obramowania pól, które nie przeszły validacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</li> <li>10b. Przejście do punktu 6.</li> </ol>
Alternatywny przebieg:	<p>Usunięcie artykułu</p> <ol style="list-style-type: none"> <li>4c. Użytkownik kliką przycisk „Usuń”,</li> <li>5c. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ten artykuł powinien zostać usunięty?”,</li> <li>6c. Użytkownik wybiera opcję „Tak”,</li> <li>7c. Serwis usuwa artykuł,</li> <li>8c. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).</li> </ol>

Alternatywny przebieg:	Anulowanie usunięcia artykułu 6ca. Użytkownik wybiera opcję „Nie” albo w obszar poza formularzem, 7ca. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).
------------------------	--

Tab. 2.14: Opis przypadku użycia - wyświetlenie listy przypisanych artykułów

Nazwa:	Wyświetlenie listy przypisanych artykułów
Cel:	Wyświetlenie listy przypisanych artykułów recenzentowi do weryfikacji
Aktorzy:	Recenzent
Warunki początkowe:	Użytkownik będący recenzentem jest zalogowany
Warunki końcowe:	Wyświetlenie listy przypisanych artykułów
Przebieg:	<ol style="list-style-type: none"> <li>Użytkownik kliką przycisk „Weryfikacja artykułów” znajdujący się w menu głównym aplikacji (Rys. 2.35),</li> <li>Aplikacja wyświetla panel z listą przypisanych użytkownikowi artykułów do weryfikacji (Rys. 2.36),</li> </ol>

Artykuły	Technologie	Weryfikacja artykułów
----------	-------------	-----------------------

Rys. 2.35: Nagłówek strony recenzenta

Źródło: opracowanie własne

Artykuły	Technologie	Weryfikacja artykułów
----------	-------------	-----------------------

**Weryfikacja artykułów**

---

Java - poradnik	Java	Język progr.		Adam Nowak	21:00, 01.01.2022
Podstawy Javy	Java	Język progr.		Kamil Dywan	20:30, 04.03.2022
React - podstawy	JS	Framework		Kamil Dywan	20:03, 07.03.2022
SQL - przykłady	JS	Język bazy danych		Kamil Dywan	20:03, 07.03.2022

---

<
1
2
...
8
>

---

Rys. 2.36: Panel z listą artykułów przypisanych recenzentowi do weryfikacji

Źródło: opracowanie własne

Tab. 2.15: Opis przypadku użycia - weryfikacja artykułu

Nazwa:	Weryfikacja przypisanych artykułów
Cel:	Wyświetlenie panelu służącego do weryfikacji artykułu
Aktorzy:	Recenzent
Warunki początkowe:	Wyświetlona lista przypisanych użytkownikowi artykułów do weryfikacji (Rys. 2.36)
Warunki końcowe:	Wyświetlenie panelu służącego do weryfikacji artykułu
Przebieg:	1. Użytkownik klikna na jeden z artykułów, 2. Serwis wyświetla formularz do przeprowadzenia weryfikacji artykułu (Rys. 2.37).

Artykuły	Technologie	Weryfikacja artykułów
----------	-------------	-----------------------

### Weryfikacja Artykułu „Podstawy Javy”

Kamil Dywan

Data utworzenia: 14.07.2022, 00:32:45

Data modyfikacji: 14.07.2022, 00:32:45

Technologia → Język programowania → Java



Dostawca: Oracle

Zastosowania:

- Backend,
- Aplikacje webowe.

## Podstawy Javy

Java jest to wspólnie używany, oparty na klasach i mocno typowany obiektowy język programowania, który może być wykorzystywany w wielu dziedzinach. Został stworzony przez zespół pod kierownictwem Jamesa Goslinga z firmy Sun Microsystems, a pierwsze wydanie tego języka odbyło się w 1996 roku. Ważną cechą tego języka jest to, że przy komplikacji źródłek kodu w nim napisanych, używana jest maszyna wirtualna nazywana JVM (ang. Java Virtual Machine).

### 1. JVM

Kod napisany w tym języku jest komplikowany do kodu bajtowego, który następnie jest wykonywany przez maszynę wirtualną. Dodatkowo używana maszyna wirtualna sprawia, że język ten jest wieloplatformowy.

...

### 5. Zastosowania

Z powodu na używaną maszynę wirtualną, programy napisane w Javie mogą być uruchomione na praktycznie wszystkich urządzeniach, co prowadzi do tego, że zastosowania tego języka mogą być bardzo rozległe. Przed wszystkim język jest wykorzystywany w aplikacjach Webowych jako backend oraz aplikacjach mobilnych. W przypadku zastosowań biznesowych język jest wykorzystywany przed wszystkim w rozwiązaniach wysokopoziomowych przez korporacje i duże firmy. Język ten z powodu używanej maszyny wirtualnej jest wolniejszy niż np. C++, dlatego też jest najczęściej wykorzystywany w aplikacjach wysokopoziomowych.

Platforma

Mobilne

Web

Wysokopoziomowe rozwiązania

Korporacyjne aplikacje

### 6. Wersje

1.0 - 1996

2.0 - grudzień 1998 - 1999

...

9.0 - wrzesień 2017

11.0 - wrzesień 2018

15.0 - wrzesień 2020

17.0 - wrzesień 2021

### 7. Linki

[1] <https://docs.oracle.com/en/java/> - oficjalna dokumentacja

[Akceptuj](#) [Odrzuć](#)

Rys. 2.37: Panel służący do weryfikacji artykułu

Źródło: opracowanie własne

Tab. 2.16: Opis przypadku użycia - akceptacja artykułu

Nazwa:	Akceptacja artykułu
Cel:	Akceptacja artykułu, co będzie skutkowało tym, że artykuł zostanie opublikowany w serwisie

Aktorzy:	Recenzent
Warunki początkowe:	Wyświetlony formularz służący do weryfikacji artykułu (Rys. 2.37)
Warunki końcowe:	Akceptacja artykułu
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik przegląda cały artykuł,</li> <li>2. Użytkownik podaje wiadomość zwrotną dotyczącą treści artykułu: uzasadnienie akceptacji artykułu oraz ewentualne wskazówki, co można by było poprawić w artykule (Rys. 2.38),</li> <li>3. Użytkownik kliką przycisk „Akceptuj”,</li> <li>4. Artykuł zostaje zaakceptowany.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje akceptację artykułu</p> <ol style="list-style-type: none"> <li>2a. Użytkownik kliką przycisk „Anuluj”,</li> <li>3a. Serwis wyłącza formularz do akceptacji artykułu.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie uzupełnia pola tekstowego</p> <ol style="list-style-type: none"> <li>2b. Przejście do punktu 3.</li> </ol>

Wiadomość zwrotna dotycząca treści artykułu

Artykuł dobry, lecz przydałoby się dodać trochę informacji o twórcach tej technologii.

Akceptuj
Odrzuć

Anuluj

Rys. 2.38: Formularz akceptacji artykułu z podaną informacją zwrotną

Źródło: opracowanie własne

Tab. 2.17: Opis przypadku użycia - odrzucenie artykułu

Nazwa:	Odrzucenie artykułu
--------	---------------------

Cel:	Odrzucenie artykułu, co będzie skutkowało tym, że artykuł nie zostanie opublikowany w serwisie, gdyż wymaga poprawek
Aktorzy:	Recenzent
Warunki początkowe:	Wyświetlony formularz służący do weryfikacji artykułu (Rys. 2.37)
Warunki końcowe:	Odrzucenie artykułu
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik przegląda cały artykuł,</li> <li>2. Użytkownik uzupełnia uzasadnienie odrzucenia artykułu (Rys. 2.39),</li> <li>3. Użytkownik kliką przycisk „Odrzuć”,</li> <li>4. Serwis sprawdza, czy użytkownik uzupełnił pole tekstowe,</li> <li>5. Po pozytywnej weryfikacji, artykuł zostaje odrzucony.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje akceptację artykułu</p> <ol style="list-style-type: none"> <li>2a. Użytkownik kliką przycisk „Anuluj”,</li> <li>3a. Serwis wyłącza formularz do odrzucenia artykułu.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie uzupełnił pola tekstowego</p> <ol style="list-style-type: none"> <li>5b. Serwis ustawia obramowanie niezupełnionego pola tekstowego na czerwono oraz umieszcza pod tym polem komunikat o konieczności jego uzupełnienia,</li> <li>6b. Powrót do punktu 2.</li> </ol>

Wiadomość zwrotna dotycząca treści artykułu

Artykuł posiada informacje nieprawdziwe odnośnie wersji technologii oraz naniesionych zmian w ramach tych wersji

Akceptuj
Odrzuć
  
Anuluj

Rys. 2.39: Formularz odrzucenia artykułu

Źródło: opracowanie własne

Tab. 2.18: Opis przypadku użycia - zarządzanie artykułami

Nazwa:	Zarządzanie artykułami
Cel:	Modyfikacja lub usunięcie artykułu
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlona lista z wyszukanymi artykułami (Rys. 2.31)
Warunki końcowe:	Modyfikacja lub usunięcie artykułu
Przebieg:	<p>Edytowanie artykułu</p> <ol style="list-style-type: none"> <li>1. Użytkownik klika na jeden z wyszukanych artykułów,</li> <li>2. Aplikacja wyświetla zawartość wybranego artykułu (Rys. 2.40),</li> <li>3. Użytkownik klika przycisk „Edytuj”,</li> <li>4. Serwis wyświetla formularz do edycji artykułu (Rys. 2.34)</li> <li>5. Użytkownik wypełnia wszystkie pola w formularzu,</li> <li>6. Użytkownik klika przycisk „Zapisz”,</li> <li>7. Aplikacja przeprowadza validację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>8. Po pozytywnej walidacji danych, aplikacja edytuje artykuł według zmian wprowadzonych przez użytkownika,</li> <li>9. Aplikacja przedstawia użytkownikowi wygląd zmienionego artykułu.</li> </ol>

Alternatywny przebieg:	<p>Użytkownik anuluje edytowanie artykułu</p> <p>5a. Użytkownik kliką przycisk „Anuluj”,      6a. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).</p>
Alternatywny przebieg:	<p>Użytkownik nie uzupełnia wszystkich pól formularza podczas edytowania artykułu</p> <p>8b. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,      9b. Przejście do punktu 5.</p>
Alternatywny przebieg:	<p>Usunięcie artykułu</p> <p>3c. Użytkownik kliką przycisk „Usuń”,      4c. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ten artykuł powinien zostać usunięty?”,      5c. Użytkownik wybiera opcję „Tak”,      6c. Serwis usuwa artykuł,      7c. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).</p>
Alternatywny przebieg:	<p>Anulowanie usunięcia artykułu</p> <p>5ca. Użytkownik wybiera opcję „Nie” albo kilka w obszarze poza formularzem,      6ca. Artykuł nie zostaje usunięty.</p>

Artykuły	Technologie	Zarządzanie użytkownikami
<p>Kamil Dywan</p> <p>Data utworzenia: 14.07.2022, 00:32:45</p> <p>Data modyfikacji: 14.07.2022, 00:32:45</p> <p>Technologia → Język programowania → Java</p>  <p>Dostawca: Oracle</p> <p>Opis:</p> <p>Java jest to współbieżny, obiektowy język programowania. Język ten jest niezwykle popularny oraz uniwersalny i dlatego trudnym zadaniem byłoby znalezienie dziedziny na pograniczu informatyki, w której problemach nie można by było wykorzystać Javy. Jednym z powodów uniewersalności tego języka jest to, że programy napisane w Javie są uruchamiane na maszynie wirtualnej, co też powoduje, że aby uruchomić program napisany w Javie, wystarczy posiadać zaistalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.</p> <h3>Podstawy Javy</h3> <p>Java jest to współbieżny, oparty na klasach i mocno typowany obiektowy język programowania, który może być wykorzystywany w wielu dziedzinach. Został stworzony przez zespół pod kierownictwem Jamesa Goslinga z firmy Sun Microsystems, a pierwsze wydanie tego języka odbyło się w 1996 roku. Ważną cechą tego języka jest to, że przy komplikacji źródeł kodu w nim napisanych, używana jest maszyna wirtualna nazywana JVM (ang. Java Virtual Machine).</p> <p>1. JVM</p> <p>Kod napisany w tym języku jest kompilowany do kodu bajtowego, który następnie jest wykonywany przez maszynę wirtualną. Dodatkowo używana maszyna wirtualna sprawia, że język ten jest wieloplatformowy.</p> <p>...</p> <p>5. Zastosowania</p> <p>Z powodu na używaną maszynę wirtualną, programy napisane w Javie mogą być uruchomione na praktycznie wszystkich urządzeniach, co prowadzi do tego, że zastosowania tego języka mogą być bardzo rozległe. Przede wszystkim język jest wykorzystywany w aplikacjach Webowych jako backend oraz aplikacjach mobilnych. W przypadku zastosowań biznesowych język jest wykorzystywany przede wszystkim w rozwiązańach wysokopoziomowych przez korporacje i duże firmy. Język ten z powodu używanej maszyny wirtualnej jest wolniejszy niż np. C++, dlatego też jest najczęściej wykorzystywany w aplikacjach wysokopoziomowych.</p> <p>Platforma Mobilne Web Wysokopoziomowe rozwiązania Korporacyjne aplikacje 6. Wersje</p> <p>1.0 - 1996 2.0 - grudzień 1998 - 1999 ... 9.0 - wrzesień 2017 11.0 - wrzesień 2018 15.0 - wrzesień 2020 17.0 - wrzesień 2021 7. Linki</p> <p>[1] <a href="https://docs.oracle.com/en/java/">https://docs.oracle.com/en/java/</a> - oficjalna dokumentacja</p> <p> 75%</p> <p><a href="#">Komentarze</a> <a href="#">Opinie</a> <a href="#">Opis technologii</a></p> <hr/> <p><a href="#">Dodaj komentarz</a></p> <p> Kamil Dywan 15.07.2022, 00:32:45 :</p> <p>Dobry artykuł</p> <p><a href="#">Dodaj komentarz</a></p> <p>v Odpowiedzi</p> <p> Adam Nowak 18.07.2022, 00:32:45 :</p> <p>Zgadzam się</p> <p><a href="#">Dodaj komentarz</a></p> <p> Kamil Dywan 20.07.2022, 00:32:45 :</p> <p>Artykuł dobry, ale przydałoby się zamieścić więcej informacji o wersjach Javy oraz opis zmian w ramach tych wersji</p> <p><a href="#">Dodaj komentarz</a></p>	<p>Status: <b>Opublikowany</b></p> <p><a href="#">Edytuj</a> <a href="#">Usuń</a></p>	

Tab. 2.19: Opis przypadku użycia - przeglądanie komentarzy o artykule

Nazwa:	Przeglądanie komentarzy o artykule
Cel:	Wyświetlenie komentarzy dotyczących danego artykułu
Aktorzy:	Użytkownik
Warunki początkowe:	Wyświetlona zawartość artykułu (Rys. 2.33)
Warunki końcowe:	Wyświetlenie komentarzy dotyczących danego artykułu
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik przewija artykuł na sam dół,</li> <li>2. Użytkownik kliką przycisk „Komentarze”,</li> <li>3. Wyświetlenie komentarzy o artykule (Rys. 2.41).</li> </ol>

Komentarze      Opinie      Opis technologii

Dodaj komentarz

Kamil Dywan 15.07.2022, 00:32:45 :

Dobry artykuł

Dodaj komentarz

v Odpowiedzi

Adam Nowak 18.07.2022, 00:32:45

Zgadzam się

Dodaj komentarz

Kamil Dywan 20.07.2022, 00:32:45 :

Artykuł dobry, ale przydałoby się zamieścić więcej informacji o wersjach Javy oraz opis zmian w ramach tych wersji

Dodaj komentarz

Rys. 2.41: Panel z listą komentarzy o artykule

Źródło: opracowanie własne

Tab. 2.20: Opis przypadku użycia - dodanie komentarza o artykule

Nazwa:	Dodanie komentarza o artykule
Cel:	Dodanie komentarza do artykułu
Aktorzy:	Zalogowany użytkownik
Warunki początkowe:	Wyświetlone komentarze o artykule (Rys. 2.41)
Warunki końcowe:	Dodanie komentarza do artykułu
Przebieg:	<p>Dodanie komentarza</p> <ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Dodaj komentarz”,</li> <li>2. Aplikacja wyświetla formularz do utworzenia komentarza,</li> <li>3. Użytkownik dodaje tekst komentarza (Rys. 2.42),</li> <li>4. Użytkownik kliką przycisk „Zapisz”,</li> <li>5. Aplikacja sprawdza, czy użytkownik podał tekst komentarza,</li> <li>6. Po pozytywnej walidacji komentarza, aplikacja dodaje komentarz do artykułu.</li> </ol>

Alternatywny przebieg:	Użytkownik anuluje tworzenie komentarza 3a. Użytkownik kliką przycisk „Anuluj” na formularzu do tworzenia komentarza, 4a. Wyłączenie formularza do dodania komentarza.
Alternatywny przebieg:	Użytkownik nie uzupełnia tekstu przy tworzeniu komentarza 6b. Aplikacja wyświetla użytkownikowi informacje o tym, że komentarz nie może być pusty, 7b. Przejście do punktu 3.

Podoba mi się ten artykuł. Dodałbyś może coś o Spring'u?

Zapisz
Anuluj

Rys. 2.42: Formularz dodania lub edytowania komentarza o artykule

Źródło: opracowanie własne

Tab. 2.21: Opis przypadku użycia - zarządzanie własnym komentarzem o artykule

Nazwa:	Zarządzanie własnym komentarzem o artykule
Cel:	Modyfikacja lub usunięcie własnego komentarza do artykułu
Aktorzy:	Zalogowany użytkownik
Warunki początkowe:	Wyświetlone komentarze o artykule (Rys. 2.41)
Warunki końcowe:	Modyfikacja lub usunięcie komentarza do artykułu
Przebieg:	<p>Edytowanie komentarza</p> <ol style="list-style-type: none"> <li>1. Użytkownik kliką trzy kropki przy jednym z komentarzy,</li> <li>2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranego komentarza (Rys. 2.43),</li> <li>3. Użytkownik kliką przycisk „Edytuj”,</li> <li>4. Aplikacja wyświetla formularz do edytowania komentarza (Rys. 2.42),</li> <li>5. Użytkownik edytuje tekst komentarza,</li> <li>6. Użytkownik kliką przycisk „Zapisz”,</li> <li>7. Aplikacja sprawdza, czy użytkownik podał tekst komentarza,</li> <li>8. Po pozytywnej walidacji komentarza, aplikacja edytuje komentarz do artykułu.</li> </ol>

Alternatywny przebieg:	Użytkownik anuluje edytowanie komentarza 5a. Użytkownik kliką przycisk „Anuluj” na formularzu do edytowania komentarza, 6a. Wyłączenie formularza do edytowania komentarza.
Alternatywny przebieg:	Użytkownik nie uzupełnia tekstu przy edytowaniu komentarza 8b. Aplikacja wyświetla użytkownikowi informacje o tym, że komentarz nie może być pusty, 9b. Przejście do punktu 5.
Alternatywny Przebieg:	Usunięcie komentarza 3c. Użytkownik kliką przycisk „Usuń” przy jednym z komentarzy, 4c. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ten komentarz powinien zostać usunięty?”, 5c. Użytkownik wybiera opcję „Tak”, 6c. Serwis usuwa komentarz.
Alternatywny przebieg:	Użytkownik anuluje usuwanie komentarza 5ca. Użytkownik wybiera opcję „Nie”, 6ca. Komentarz nie zostaje usunięty.



Rys. 2.43: Dostępne opcje do zarządzania komentarzem

Źródło: opracowanie własne

Tab. 2.22: Opis przypadku użycia - usuwanie komentarzy o artykule

Nazwa:	Usuwanie komentarzy o artykule
Cel:	Usunięcie wybranego komentarza o artykule
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlone komentarze o artykule (Rys. 2.44)
Warunki końcowe:	Usunięcie komentarza do artykułu

Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką trzy kropki przy jednym z komentarzy,</li> <li>2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranego komentarza (Rys. 2.43),</li> <li>3. Użytkownik kliką przycisk „Usuń”,</li> <li>4. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ten komentarz powinien zostać usunięty?”,</li> <li>5. Użytkownik wybiera opcję „Tak”,</li> <li>6. Serwis usuwa komentarz.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje usuwanie komentarza</p> <ol style="list-style-type: none"> <li>5a. Użytkownik wybiera opcję „Nie”,</li> <li>6a. Komentarz nie zostaje usunięty.</li> </ol>

Rys. 2.44: Lista komentarzy o artykule widziana z perspektywy administratora

Źródło: opracowanie własne

Tab. 2.23: Opis przypadku użycia - przeglądanie opinii o artykule

Nazwa:	Przeglądanie opinii o artykule
Cel:	Wyświetlenie opinii dotyczących danego artykułu
Aktorzy:	Użytkownik
Warunki początkowe:	Wyświetlona zawartość artykułu (Rys. 2.33)
Warunki końcowe:	Wyświetlenie opinii dotyczących danego artykułu
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik przewija artykuł na sam dół,</li> <li>2. Użytkownik kliką przycisk „Opinie”,</li> <li>3. Aplikacja wyświetla opinie o artykule (Rys. 2.45).</li> </ol>

Komentarze    Opinie    Opis technologii

[Dodaj opinię](#)

★★★★★

Kamil Dywan 15.07.2022, 00:32:45 :

Bardzo dobry artykuł

Like 2    Dislike X

★★★★★☆

Adam Nowak 20.07.2022, 00:32:45

Artykuł dobry, ale przydałoby się zamieścić więcej informacji o wersjach Javy oraz opis zmian w ramach tych wersji

Like 10    Dislike 1

Rys. 2.45: Panel z listą opinii o artykule

Źródło: opracowanie własne

Tab. 2.24: Opis przypadku użycia - dodanie opinii o artykule

Nazwa:	Dodanie opinii o artykule
Cel:	Dodanie opinii o artykule
Aktorzy:	Zalogowany użytkownik, który nie jest autorem artykułu
Warunki początkowe:	Wyświetlone opinie o artykule (Rys. 2.47)
Warunki końcowe:	Dodanie opinii o artykule
Przebieg:	<p>Dodanie opinii</p> <ol style="list-style-type: none"> <li>Użytkownik klika przycisk „Dodaj opinię”,</li> <li>Aplikacja wyświetla formularz do utworzenia opinii,</li> <li>Użytkownik dodaje tekst opinii oraz wystawia ocenę (Rys. 2.46),</li> <li>Użytkownik klika przycisk „Zapisz”,</li> <li>Aplikacja sprawdza, czy użytkownik podał tekst opinii oraz ocenę,</li> <li>Po pozytywnej walidacji, aplikacja dodaje opinię.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje tworzenie opinii</p> <ol style="list-style-type: none"> <li>Użytkownik klika przycisk „Anuluj” na formularzu do tworzenia opinii,</li> <li>Wyłączenie formularza do dodania opinii.</li> </ol>

Alternatywny przebieg:	<p>Użytkownik nie uzupełnia wszystkich wymaganych pól formularza</p> <p>6b. Aplikacja zmienia kolor obramowania nieuzupełnionych pól na czerwono oraz wyświetla pod tymi polami komunikat „Pole wymagane”,            7b. Przejście do punktu 3.</p>
------------------------	--



Generalnie artykuł w porządku, lecz brakuje trochę informacji o wersjach oraz naniesionych zmianach w ramach tych wersji

Zapisz

Anuluj

Rys. 2.46: Uzupełniony przykładowymi danymi formularz dodania lub edytowania opinii o artykule

Źródło: opracowanie własne

Tab. 2.25: Opis przypadku użycia - zarządzanie własną opinią o artykule

Nazwa: Cel: Aktorzy: Warunki początkowe: Warunki końcowe:  Przebieg:  Alternatywny przebieg:	Zarządzanie własną opinią o artykule Modyfikacja lub usunięcie własnej opinii do artykułu Autor opinii Wyświetlone opinie o artykule (Rys. 2.47) Modyfikacja lub usunięcie własnej opinii do artykułu  Edytowanie opinii 1. Użytkownik kliką trzy kropki przy jednej z opinii, 2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranej opinii 3. Użytkownik kliką przycisk „Edytuj”, 4. Aplikacja wyświetla formularz do edytowania opinii (Rys. 2.46), 5. Użytkownik edytuje opinię, 6. Użytkownik kliką przycisk „Zapisz”, 7. Aplikacja sprawdza, czy użytkownik podał tekst opinię oraz ocenę, 8. Po pozytywnej walidacji, aplikacja edytuje opinię.  Użytkownik anuluje edytowanie opinii 5a. Użytkownik kliką przycisk „Anuluj” na formularzu do edytowania opinii, 6a. Wyłączenie formularza do edytowania opinii.
--	---

Alternatywny przebieg:	Użytkownik nie uzupełnia tekstu przy edytowaniu opinii 8b. Aplikacja zmienia kolor obramowania nieuzupełnionych pól na czerwono oraz wyświetla pod tymi polami komunikat „Pole wymagane”, 9b. Przejście do punktu 5.
Alternatywny Przebieg:	Usunięcie opinii 3c. Użytkownik kliką przycisk „Usuń”, 4c. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ta opinia powinna zostać usunięta?”, 5c. Użytkownik wybiera opcję „Tak”, 6c. Serwis usuwa opinię.
Alternatywny przebieg:	Użytkownik anuluje usuwanie opinii 5ca. Użytkownik wybiera opcję „Nie”, 6ca. Opinia nie zostaje usunięta.

Tab. 2.26: Opis przypadku użycia - usuwanie opinii o artykule

Nazwa:	Usuwanie komentarzy o artykule
Cel:	Usunięcie wybranej opinii o artykule
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlone opinie o artykule (Rys. 2.47)
Warunki końcowe:	Usunięcie opinii do artykułu
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką trzy kropki przy jednej z opinii,</li> <li>2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranej opinii</li> <li>3. Użytkownik kliką przycisk „Usuń”</li> <li>4. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ta opinia powinna zostać usunięta?”,</li> <li>5. Użytkownik wybiera opcję „Tak”,</li> <li>6. Serwis usuwa opinię.</li> </ol>
Alternatywny przebieg:	Użytkownik anuluje usuwanie opinii 5a. Użytkownik wybiera opcję „Nie”, 6a. Opinia nie zostaje usunięta.

Komentarze    Opinie    Opis technologii

---

[Dodaj opinię](#)

★★★★★

Kamil Dywan 15.07.2022, 00:32:45 :

Bardzo dobry artykuł

Like 2    Report X

★★★★★

Adam Nowak 20.07.2022, 00:32:45 :

Artykuł dobry, ale przydałoby się zamieścić więcej informacji o wersjach Javy oraz opis zmian w ramach tych wersji

Like 10    Report 1

---

Rys. 2.47: Panel z listą opinii o artykule widziany z perspektywy administratora

Źródło: opracowanie własne

Tab. 2.27: Opis przypadku użycia - dodanie akceptacji opinii o artykule

Nazwa:	Dodanie akceptacji opinii o artykule
Cel:	Dodanie akceptacji opinii o artykule
Aktorzy:	Zalogowany użytkownik, który nie jest autorem opinii o artykule oraz autorem akceptacji opinii o artykule
Warunki początkowe:	Wyświetlone opinie o artykule (Rys. 2.47)
Warunki końcowe:	Wyświetlenie opinii z zaktualizowanym stanem akceptacji pozytywnych i negatywnych
Przebieg:	<p>Dodanie kciuka w górę</p> <ol style="list-style-type: none"> <li>Użytkownik kliką przy jednej z opinii kciuk w góre,</li> <li>Aplikacja zwiększa o 1 liczbę pozytywnych akceptacji wybranej opinii,</li> <li>Aplikacja zmienia kolor kciuka w góre na czarny,</li> <li>Aplikacja wyświetla opinię z zaktualizowanym stanem akceptacji pozytywnych i negatywnych.</li> </ol>
Alternatywny przebieg:	<p>Dodanie kciuka w dół</p> <ol style="list-style-type: none"> <li>a. Użytkownik kliką przy jednej z opinii kciuk w dół,</li> <li>a. Aplikacja zwiększa o 1 liczbę negatywnych akceptacji wybranej opinii,</li> <li>a. Aplikacja zmienia kolor kciuka w dół na czarny,</li> <li>a. Aplikacja wyświetla opinię z zaktualizowanym stanem akceptacji pozytywnych i negatywnych.</li> </ol>

Tab. 2.28: Opis przypadku użycia - usunięcie akceptacji opinii o artykule

Nazwa:	Usunięcie akceptacji opinii o artykule
Cel:	Usunięcie akceptacji opinii o artykule

Aktorzy:	Autor akceptacji opinii o artykule
Warunki początkowe:	Wyświetlone opinie o artykule (Rys. 2.47)
Warunki końcowe:	Wyświetlenie opinii z aktualizowanym stanem akceptacji pozytywnych i negatywnych
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką w czerwony znak „X” przy akceptacjach wybranej opinii,</li> <li>2. Aplikacja usuwa akceptację opinii,</li> <li>3. Aplikacja zmienia kolor ustawionego kciuka na biały,</li> <li>4. Aplikacja wyświetla opinię z aktualizowanym stanem akceptacji pozytywnych i negatywnych.</li> </ol>

Tab. 2.29: Opis przypadku użycia - wyszukiwanie technologii

Nazwa:	Wyszukiwanie technologii
Opis:	Wyszukiwanie technologii według podanych przez użytkownika kryteriów i wyświetlenie wyników tego wyszukiwania w postaci listy technologii
Aktorzy:	Użytkownik
Warunki początkowe:	Brak
Warunki końcowe:	Wyświetlenie listy wyszukanych technologii
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Technologie” znajdujący się w panelu nawigacji strony (Rys. 2.14),</li> <li>2. Aplikacja wyświetla panel do wyszukiwania technologii,</li> <li>3. Użytkownik uzupełnia kryteria wyszukiwania,</li> <li>4. Użytkownik kliką przycisk „Szukaj”,</li> <li>5. Serwis wyszukuje technologii,</li> <li>6. Aplikacja wyświetla listę wyszukanych technologii (Rys. 2.48).</li> </ol>

Artykuły	Technologie																																			
<h3 style="margin: 0;">Wyszukiwanie technologii</h3> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Nazwa</td> <td style="width: 85%; text-align: right;"><input type="text" value="Java"/> v</td> </tr> <tr> <td>Kategoria</td> <td style="text-align: right;"><input type="text" value="Język programowania"/> v</td> </tr> <tr> <td>Zastosowania</td> <td style="text-align: right;"><input type="text" value="Backend"/> v</td> </tr> <tr> <td>Dostawca</td> <td style="text-align: right;"><input type="text" value="Oracle"/> v</td> </tr> <tr> <td>Data powstania</td> <td style="text-align: right;"><input type="text" value="02.03.1950"/> [03.05.2022]</td> </tr> <tr> <td>Data modyfikacji</td> <td style="text-align: right;"><input type="text" value="09.07.2020"/> [28.08.2022]</td> </tr> </table> <input style="width: 100px; background-color: #0072BD; color: white; font-weight: bold; padding: 2px 10px; border: none; border-radius: 5px; margin-bottom: 5px;" type="button" value="Szukaj"/> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Sortowanie</td> <td style="width: 85%; text-align: right;"><input type="text" value=""/> v</td> </tr> <tr> <td colspan="2" style="text-align: right; padding-top: 5px;"> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">1</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">2</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">8</span> <span style="border: 1px solid black; padding: 2px 5px;">&gt;</span> </td> </tr> </table> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Java</td> <td style="width: 33%;">Języki programowania</td> <td style="width: 33%;">Oracle</td> </tr> <tr> <td>21:00, 20.08.2022</td> <td></td> <td></td> </tr> <tr> <td>C++</td> <td>Języki programowania</td> <td>20:00, 16.06.2020</td> </tr> <tr> <td>Javascript</td> <td>Języki programowania</td> <td>22:00, 21.08.2022</td> </tr> <tr> <td>React</td> <td>Framework</td> <td>23:00, 22.09.2022</td> </tr> </table> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Sortowanie</td> <td style="width: 85%; text-align: right;"><input type="text" value=""/> v</td> </tr> <tr> <td colspan="2" style="text-align: right; padding-top: 5px;"> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">1</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">2</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">8</span> <span style="border: 1px solid black; padding: 2px 5px;">&gt;</span> </td> </tr> </table>		Nazwa	<input type="text" value="Java"/> v	Kategoria	<input type="text" value="Język programowania"/> v	Zastosowania	<input type="text" value="Backend"/> v	Dostawca	<input type="text" value="Oracle"/> v	Data powstania	<input type="text" value="02.03.1950"/> [03.05.2022]	Data modyfikacji	<input type="text" value="09.07.2020"/> [28.08.2022]	Sortowanie	<input type="text" value=""/> v	<span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">1</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">2</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">8</span> <span style="border: 1px solid black; padding: 2px 5px;">&gt;</span>		Java	Języki programowania	Oracle	21:00, 20.08.2022			C++	Języki programowania	20:00, 16.06.2020	Javascript	Języki programowania	22:00, 21.08.2022	React	Framework	23:00, 22.09.2022	Sortowanie	<input type="text" value=""/> v	<span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">1</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">2</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">8</span> <span style="border: 1px solid black; padding: 2px 5px;">&gt;</span>	
Nazwa	<input type="text" value="Java"/> v																																			
Kategoria	<input type="text" value="Język programowania"/> v																																			
Zastosowania	<input type="text" value="Backend"/> v																																			
Dostawca	<input type="text" value="Oracle"/> v																																			
Data powstania	<input type="text" value="02.03.1950"/> [03.05.2022]																																			
Data modyfikacji	<input type="text" value="09.07.2020"/> [28.08.2022]																																			
Sortowanie	<input type="text" value=""/> v																																			
<span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">1</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">2</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">8</span> <span style="border: 1px solid black; padding: 2px 5px;">&gt;</span>																																				
Java	Języki programowania	Oracle																																		
21:00, 20.08.2022																																				
C++	Języki programowania	20:00, 16.06.2020																																		
Javascript	Języki programowania	22:00, 21.08.2022																																		
React	Framework	23:00, 22.09.2022																																		
Sortowanie	<input type="text" value=""/> v																																			
<span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;"></span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">1</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">2</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">8</span> <span style="border: 1px solid black; padding: 2px 5px;">&gt;</span>																																				

Rys. 2.48: Panel do wyszukiwania technologii z podanymi kryteriami wyszukiwania oraz listą wyszukanych technologii

Źródło: opracowanie własne

Tab. 2.30: Opis przypadku użycia - sortowanie technologii

Nazwa:	Sortowanie technologii
Opis:	Posortowanie wyszukanych technologii według opcji sortowania wybranej przez użytkownika oraz wyświetlenie listy tych posortowanych technologii
Aktorzy:	Użytkownik
Warunki początkowe:	Wyświetlona lista wyszukanych technologii (Rys. 2.48)
Warunki końcowe:	Wyświetlenie listy posortowanych technologii
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik kliką na rozsuwaną listę „Sortowanie”,</li> <li>2. Użytkownik wybiera jedną z opcji sortowania (Rys. 2.49),</li> <li>3. Aplikacja sortuje listę technologii według wybranej opcji sortowania,</li> <li>4. Aplikacja wyświetla listę posortowanych technologii.</li> </ol>

Sortowanie	▼
Liczba opinii: malejąco	
Liczba opinii: rosnąco	
Data pierwszego wydania: malejąco	
Data pierwszego wydania: rosnąco	
Data ostatniego wydania: malejąco	
Data ostatniego wydania: rosnąco	
Średnia ocena: malejąco	
Średnia ocena: rosnąco	

Rys. 2.49: Dostępne opcje sortowania technologii

Źródło: opracowanie własne

Tab. 2.31: Opis przypadku użycia - wyświetlenie opisu technologii

Nazwa:	Wyświetlenie opisu technologii
Cel:	Wyświetlenie opisu technologii
Aktorzy:	Użytkownik
Warunki początkowe:	Brak
Warunki końcowe:	Wyświetlenie opisu technologii
Przebieg:	1. Użytkownik wyszukuje technologie (Tab. 2.29) 2. Użytkownik klikna na jedną z wyszukanych technologii, 3. Aplikacja wyświetla opis technologii (Rys. 2.50).
Alternatywny przebieg:	Użytkownik wyświetla opis technologii poprzez artykuł 1a. Użytkownik wyszukuje artykuły (Tab. 2.9), 2a. Użytkownik wyświetla zawartość artykułu (Tab. 2.11), 3a. Użytkownik klikna przycisk „Opis technologii”, 4a. Powrót do punktu 3.

Data pierwszego wydania: 01.01.1995, 12:25:32

Data ostatniego wydania: 22.03.2022, 21:32:45

Technologia → Język programowania → Java



Dostawca: Oracle

Opis:

Java jest to wspólnobieżny, obiektowy język programowania. Język ten jest niezwykle popularny oraz uniwersalny i dlatego trudnym zadaniem byłoby znalezienie dziedziny na pograniczu informatyki, w której problemach nie można by było wykorzystać Javy. Jednym z powodów uniwersalności tego języka jest to, że programy napisane w Javie są uruchamiane na maszynie wirtualnej, co też powoduje, że aby uruchomić program napisany w Javie, wystarczy posiadać zaistalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.

Rys. 2.50: Panel z zawartością technologii

Źródło: opracowanie własne

Tab. 2.32: Opis przypadku użycia - edycja klasyfikacji technologii

Nazwa:	Edycja klasyfikacji technologii
Cel:	Dodanie nowej technologii, modyfikacja lub usunięcie istniejącej technologii
Aktorzy:	Recenzent
Warunki początkowe:	Wyświetlony panel do wyszukiwania technologii (Rys. 2.52)
Warunki końcowe:	Dodanie nowej technologii, modyfikacja lub usunięcie istniejącej technologii

Przebieg:	<p>Dodawanie technologii</p> <ol style="list-style-type: none"> <li>1. Użytkownik kliką „Dodaj technologię”,</li> <li>2. Aplikacja wyświetla formularz do utworzenia technologii,</li> <li>3. Użytkownik wypełnia wszystkie pola w formularzu (Rys. 2.51)</li> <li>4. Użytkownik kliką przycisk „Zapisz”,</li> <li>5. Serwis przeprowadza validację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>6. Po pozytywnej validacji danych, aplikacja tworzy technologię,</li> <li>7. Aplikacja wyświetla użytkownikowi wygląd utworzonej technologii.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje dodanie technologii</p> <ol style="list-style-type: none"> <li>3a. Użytkownik kliką przycisk „Anuluj”,</li> <li>4a. Użytkownik zostaje przekierowany do strony z wyszukiwaniem technologii (Rys. 2.52).</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie uzupełnia wszystkich pól formularza podczas dodawania artykułu</p> <ol style="list-style-type: none"> <li>6b. Aplikacja zmienia kolor obramowania pól, które nie przeszły validacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie,</li> <li>7b. Przejście do punktu 3.</li> </ol>
Alternatywny przebieg:	<p>Edytowanie technologii</p> <ol style="list-style-type: none"> <li>1c. Użytkownik wyszukuje technologie (Tab. 2.29),</li> <li>2c. Użytkownik wyświetla opis technologii (Tab. 2.31),</li> <li>3c. Użytkownik kliką przycisk „Edytuj”,</li> <li>4c. Aplikacja wyświetla formularz do edytowania technologii,</li> <li>5c. Użytkownik wypełnia wszystkie pola w formularzu (Rys. 2.51)</li> <li>6c. Użytkownik kliką przycisk „Zapisz”,</li> <li>7c. Serwis przeprowadza validację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych,</li> <li>8c. Po pozytywnej validacji danych, aplikacja modyfikuje technologię,</li> <li>9c. Aplikacja wyświetla użytkownikowi wygląd zmodyfikowanej technologii.</li> </ol>

Alternatywny przebieg:	Użytkownik anuluje edytowanie technologii 3ca. Użytkownik kliką przycisk „Anuluj”, 4ca. Wyłączenie formularza do edytowania technologii.
Alternatywny przebieg:	Użytkownik nie uzupełnia wszystkich pól formularza podczas edytowania artykułu 8cb. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie, 9cb. Przejście do punktu 5c.
Alternatywny przebieg:	Usunięcie technologii 3cc. Użytkownik kliką przycisk „Usuń”, 4cc. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ta technologia powinna zostać usunięta?”, 5cc. Użytkownik wybiera opcję „Tak”, 6cc. Serwis usuwa wybraną technologię.
Alternatywny przebieg:	Użytkownik anuluje usunięcie technologii 5cca. Użytkownik wybiera opcję „Nie”, 6cca. Technologia nie zostaje usunięta.

Artykuły	Technologie	Weryfikacja artykułów
----------	-------------	-----------------------

Nazwa	Java *
Kategoria	Języki progr. ▼ *
Dostawca	Oracle
Data ostatniego wydania	01.01.1996
Data ostatniego wydania	22.03.2022
Ikona	<input type="button" value="ikona.png +"/>

Opis

Java jest to współbieżny, obiektowy język programowania. Język ten jest niezwykle popularny oraz uniwersalny i dlatego trudnym zadaniem byłoby znalezienie dziedziny na pograniczu informatyki, w której problemach nie można by było wykorzystać Javy. Jednym z powodów uniwersalności tego języka jest to, że programy napisane w Javie są uruchamiane na maszynie wirtualnej, co też powoduje, że aby uruchomić program napisany w Javie, wystarczy posiadać zaistalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.

Rys. 2.51: Panel z formularzem dodania lub modyfikacji technologii

Źródło: opracowanie własne

Artykuły	Technologie																
<b>Wyszukiwanie technologii</b> <span style="float: right; border: 1px solid blue; padding: 2px;">Dodaj technologię</span>																	
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Nazwa</td> <td style="width: 85%;"><input type="text" value="Java"/></td> </tr> <tr> <td>Kategoria</td> <td><input type="text" value="Język programowania"/></td> </tr> <tr> <td>Zastosowania</td> <td><input type="text" value="Backend"/></td> </tr> <tr> <td>Dostawca</td> <td><input type="text" value="Oracle"/></td> </tr> <tr> <td>Data powstania</td> <td><input type="text" value="02.03.1950"/> <input type="text" value="03.05.2022"/></td> </tr> <tr> <td>Data modyfikacji</td> <td><input type="text" value="09.07.2020"/> <input type="text" value="28.08.2022"/></td> </tr> </table>		Nazwa	<input type="text" value="Java"/>	Kategoria	<input type="text" value="Język programowania"/>	Zastosowania	<input type="text" value="Backend"/>	Dostawca	<input type="text" value="Oracle"/>	Data powstania	<input type="text" value="02.03.1950"/> <input type="text" value="03.05.2022"/>	Data modyfikacji	<input type="text" value="09.07.2020"/> <input type="text" value="28.08.2022"/>				
Nazwa	<input type="text" value="Java"/>																
Kategoria	<input type="text" value="Język programowania"/>																
Zastosowania	<input type="text" value="Backend"/>																
Dostawca	<input type="text" value="Oracle"/>																
Data powstania	<input type="text" value="02.03.1950"/> <input type="text" value="03.05.2022"/>																
Data modyfikacji	<input type="text" value="09.07.2020"/> <input type="text" value="28.08.2022"/>																
<input type="button" value="Szukaj"/>																	
Sortowanie <input style="width: 50px;" type="text"/> <span style="border: 1px solid #ccc; padding: 2px;">&lt;</span> <span style="border: 1px solid #ccc; padding: 2px;">1</span> <span style="border: 1px solid #ccc; padding: 2px;">2</span> <span style="border: 1px solid #ccc; padding: 2px;">...</span> <span style="border: 1px solid #ccc; padding: 2px;">8</span> <span style="border: 1px solid #ccc; padding: 2px;">&gt;</span>																	
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Java</td> <td style="width: 25%;">Języki programowania</td> <td style="width: 25%;">Oracle</td> <td style="width: 25%;">21:00, 20.08.2022</td> </tr> <tr> <td>C++</td> <td>Języki programowania</td> <td></td> <td>20:00, 16.06.2020</td> </tr> <tr> <td>Javascript</td> <td>Języki programowania</td> <td></td> <td>22:00, 21.08.2022</td> </tr> <tr> <td>React</td> <td>Framework</td> <td></td> <td>23:00, 22.09.2022</td> </tr> </table>		Java	Języki programowania	Oracle	21:00, 20.08.2022	C++	Języki programowania		20:00, 16.06.2020	Javascript	Języki programowania		22:00, 21.08.2022	React	Framework		23:00, 22.09.2022
Java	Języki programowania	Oracle	21:00, 20.08.2022														
C++	Języki programowania		20:00, 16.06.2020														
Javascript	Języki programowania		22:00, 21.08.2022														
React	Framework		23:00, 22.09.2022														
Sortowanie <input style="width: 50px;" type="text"/> <span style="border: 1px solid #ccc; padding: 2px;">&lt;</span> <span style="border: 1px solid #ccc; padding: 2px;">1</span> <span style="border: 1px solid #ccc; padding: 2px;">2</span> <span style="border: 1px solid #ccc; padding: 2px;">...</span> <span style="border: 1px solid #ccc; padding: 2px;">8</span> <span style="border: 1px solid #ccc; padding: 2px;">&gt;</span>																	

Rys. 2.52: Panel do wyszukiwania technologii z podanymi kryteriami wyszukiwania i listą technologii widziany z perspektywy recenzenta

Źródło: opracowanie własne

Tab. 2.33: Opis przypadku użycia - wyszukiwanie artykułów z przeterminowaną weryfikacją

Nazwa:	Wyszukiwanie artykułów z przeterminowaną weryfikacją
Cel:	Wyszukanie artykułów z przeterminowaną weryfikacją
Aktorzy:	System
Warunki początkowe:	Minęła godzina od ostatniego wyszukania artykułów z przeterminowaną weryfikacją
Warunki końcowe:	Zwrócenie artykułów z przeterminowaną weryfikacją
Przebieg:	<ol style="list-style-type: none"> <li>1. Sprawdzenie artykułów, które artykuły mają przedawnioną weryfikację,</li> <li>2. Zapisanie artykułów z przedawnioną weryfikacją do kolejki artykułów do weryfikacji (jeśli nie są jeszcze w tej kolejce),</li> <li>3. Zwrócenie artykułów znajdujących się w kolejce artykułów do weryfikacji.</li> </ol>

Tab. 2.34: Opis przypadku użycia - przypisanie recenzentowi artykułu do weryfikacji

Nazwa:	Przypisanie recenzentowi artykułu do weryfikacji
Cel:	Przypisanie recenzentowi artykułu do weryfikacji
Aktorzy:	System
Warunki początkowe:	Zalogowany użytkownik utworzył artykuł lub autor artykułu dokonał modyfikacji istniejącego artykułu
Warunki końcowe:	Przypisanie wybranemu recenzentowi artykułu do weryfikacji lub oczekiwanie na dostępnego recenzenta
Przebieg:	<p>Jest przynajmniej jeden dostępny recenzent</p> <p>1. Przypisanie wybranemu recenzentowi artykułu do weryfikacji (Artykuły są przypisywane recenzentom na podstawie ich dostępności, posiadanych przez nich list technologii, w których są ekspertami oraz liczby zrecenzowanych przez nich artykułów. W pierwszej kolejności nie będą brani pod uwagę niedostępni recenzenci, następnie będą wybierani recenzenci, którzy są ekspertami w jak najbardziej szczegółowej technologii (np. Java jest bardziej szczegółowa niż języki programowania), której dotyczy artykuł oraz w trzeciej kolejności będą wybierani recenzenci, którzy zrecenzowali najmniejszą liczbę artykułów).</p>
Alternatywny przebieg:	<p>Nie ma dostępnych recenzentów</p> <p>1a. System zapisuje artykuł do kolejki artykułów do weryfikacji,</p> <p>2a. System czeka 1 godzinę na pojawienie się co najmniej 1 dostępnego recenzenta,</p> <p>3a. Nie ma dalej dostępnych recenzentów.</p>
Alternatywny przebieg:	<p>Pojawił się dostępny recenzent</p> <p>3aa. Powrót do punktu 1.</p>

Tab. 2.35: Opis przypadku użycia - przypisanie artykułu do weryfikacji innemu recenzentowi

Nazwa:	Przypisanie artykułu do weryfikacji innemu recenzentowi
Cel:	Przypisanie artykułu do weryfikacji innemu recenzentowi
Aktorzy:	System
Warunki początkowe:	Zalogowany użytkownik utworzył artykuł lub autor artykułu dokonał modyfikacji istniejącego artykułu
Warunki końcowe:	Przypisanie wybranemu recenzentowi artykułu do weryfikacji lub oczekiwanie na dostępnego recenzenta
Przebieg:	<p>1. Mija termin weryfikacji artykułu,</p> <p>2. Przypisanie artykułu do weryfikacji innemu recenzentowi (Tab 2.34).</p>

Tab. 2.36: Opis przypadku użycia - wyszukiwanie technologii, w których recenzent jest ekspertem

Nazwa:	Wyszukiwanie technologii, w których recenzent jest ekspertem
Cel:	Wyszukiwanie technologii, w których recenzent jest ekspertem oraz wyświetlenie tych technologii
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlony panel z danymi wybranego recenzenta (Rys 2.53)
Warunki końcowe:	Wyświetlenie technologii, w których recenzent jest ekspertem
Przebieg:	<ol style="list-style-type: none"> <li>1. Użytkownik wprowadza kryteria wyszukiwania,</li> <li>2. Aplikacja wyświetla wyszukane technologie (Rys 2.53).</li> </ol>

IT Tech Technologie IT Kamil Dywan

Artykuły	Technologie	Zarządzanie użytkownikami
----------	-------------	---------------------------

### Dane użytkownika

Imię	<input type="text" value="Kamil"/>
Nazwisko	<input type="text" value="Dywan"/>
Nazwa użytkownika	<input type="text" value="kamil.dywan"/>
Pseudonim	<input type="text" value="kamil_dywan"/>
E-mail	<input type="text" value="kamil.dywan@mail.com"/>

[Reset hasła](#)

[Zapisz](#)
[Anuluj](#)

### Role użytkownika

Recenzent	<input checked="" type="checkbox"/>
Administrator	<input type="checkbox"/>

### Znane przez użytkownika technologie

<a href="#">Dodaj</a>		<a href="#">Wyszukaj</a> <input type="text" value="a"/>
Java	Języki programowania	<a href="#">Usuń</a>
Angular	Framework	<a href="#">Usuń</a>
Javascript	Języki programowania	<a href="#">Usuń</a>
React	Framework	<a href="#">Usuń</a>

< 1 2 ... 8 >

Rys. 2.53: Panel do wyszukiwania technologii, w których recenzent jest ekspertem

Źródło: opracowanie własne

Tab. 2.37: Opis przypadku użycia - zarządzanie listą technologii, w których recenzent jest ekspertem

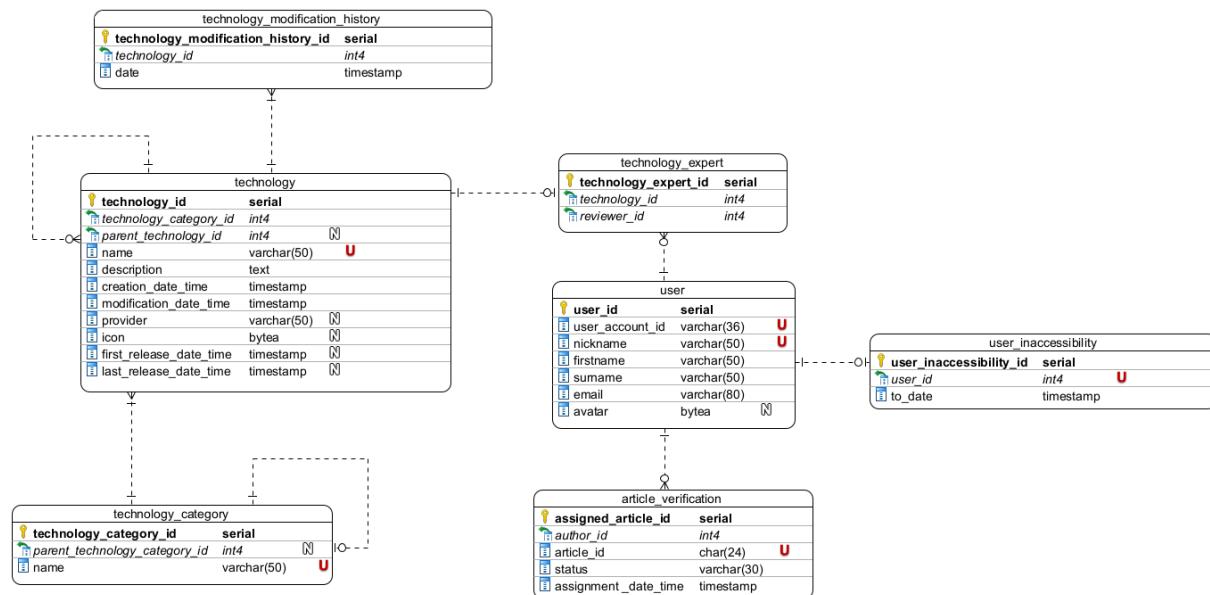
Nazwa:	Zarządzanie listą technologii, w których recenzent jest ekspertem
Cel:	Przypisanie lub usunięcie technologii, w których recenzent jest ekspertem
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlony panel z wyszukanymi technologiami, w których recenzent jest ekspertem (Rys 2.53)
Warunki końcowe:	Wyświetlenie zaktualizowanej listy technologii recenzenta
Przebieg:	<p>Przypisanie technologii</p> <ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Dodaj”,</li> <li>2. Aplikacja wyświetla formularz z dodaniem technologii,</li> <li>3. Użytkownik wybiera technologię (Rys 2.54),</li> <li>4. Użytkownik kliką przycisk „Zatwierdź”,</li> <li>5. Serwis sprawdza, czy użytkownik podał technologię,</li> <li>6. Użytkownik podał technologię, zatem aplikacja przypisuje technologię recenzentowi,</li> <li>7. Aplikacja wyświetla zaktualizowane technologie.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik anuluje dodanie technologii</p> <ol style="list-style-type: none"> <li>3a. Użytkownik kliką przycisk „Anuluj” lub kliką w obszar poza formularzem,</li> <li>4a. Wyłączenie formularza dodania technologii.</li> </ol>
Alternatywny przebieg:	<p>Użytkownik nie podał technologii</p> <ol style="list-style-type: none"> <li>6b. Aplikacja wyświetla komunikat o konieczności podania technologii,</li> <li>7b. Powrót do punktu 2.</li> </ol>
Alternatywny przebieg:	<p>Usunięcie przypisania technologii</p> <ol style="list-style-type: none"> <li>1c. Użytkownik kliką przy jednej z technologii przycisk „Usuń”</li> <li>2c. Aplikacja usuwa recenzentowi przypisanie wybranej technologii,</li> <li>3c. Powrót do punktu 7.</li> </ol>

Rys. 2.54: Formularz przypisania technologii do recenzenta

Źródło: opracowanie własne

## 2.5. Projekt baz danych

Na podstawie zdefiniowanych wymagań funkcjonalnych oraz szczegółowych opisów przypadków użycia, zaprojektowano diagram związków encji (ERD) dla głównej bazy danych (obiektowo-relacyjna baza) oraz utworzono schematy danych dla bazy artykułów (nierelacyjna dokumentowa baza). Do sporządzenia diagramu ERD wykorzystano narzędzie Visual Paradigm Standard. Diagram ERD dla głównej bazy danych przedstawiono na rysunku 2.55.



Rys. 2.55: Diagram ERD dla głównej bazy danych

Źródło: opracowanie własne

Dla bazy danych artykułów utworzono osobne schematy danych w postaci formatu JSON dla następujących kolekcji: artykuł, komentarz oraz opinia. Nie zdecydowano się na tylko jedną kolekcję zawierającą te wszystkie informacje, gdyż założono, że użytkownik będzie mógł w jednym momencie przeglądać albo komentarze albo opinie i nie ma potrzeby pobierania razem obu tych informacji. Dla zdefiniowanych schematów danych bazy artykułów utworzono również po jednym przykładowym dokumencie w formacie JSON.

Kod źródłowy schematu artykułów znajduje się na listingu 2.1.

Listing 2.1: Schemat kolekcji artykułów w postaci JSON dla bazy danych artykułów

```

1  {
2      "$jsonSchema": {
3          "bsonType": "object",
4          "title": "articles",
5          "required": [
6              "authorId",
7              "technologyId",
8              "title",
9              "content",
10             "status",
11             "creationDate",
12             "modificationDate"
13         ],
14         "properties": {
15             "_id": {
16                 "bsonType": "objectId"
17             },
18             "authorId": {
19                 "bsonType": "int"
20             },
21             "technologyId": {
22                 "bsonType": "int"
23             },
24             "title": {
25                 "bsonType": "string",
26                 "uniqueItems": true
27             },
28             "content": {
29                 "bsonType": "string"
30             },
31             "status": {
32                 "bsonType": "string",
33                 "enum": [
34                     "NEW",
35                     "ASSIGNING_TO_VERIFICATION",
36                     "VERIFICATION",
37                     "EXPIRED_VERIFICATION",
38                     "REFUSED",
39                     "PUBLISHED"
40                 ]
41             },
42             "creationDate": {
43                 "bsonType": "date"
44             },
45             "modificationDate": {
46                 "bsonType": "date"
47             },
48             "averageRating": {
49                 "bsonType": "double",
50                 "minimum": 0
51             }
52         }
53     }
54 }
```

Na podstawie schematu artykułów (listing 2.1) utworzono przykładowy dokument, który jest przedstawiony na listingu 2.2

Listing 2.2: Przykładowy dokument dla kolekcji artykułów zgodny ze zdefiniowanym schematem

```

1  {
2      "_id": {
3          "$oid": "634308ab8f35533467caa4f7"
4      },
5      "authorId": 27,
6      "technologyId": 1,
7      "title": "Podstawy Javy1",
8      "content": "<p style=\"text-align: justify;\">Java jest to wsp&oacute;wczesny, oparty na klasach i mocno typowany obiektowy język programowania...",
9      "status": "NEW",
10     "creationDate": {
11         "$date": {
12             "$numberLong": "1665337515291"
13         }
14     },
15     "modificationDate": {
16         "$date": {
17             "$numberLong": "1665337515294"
18         }
19     }
20 }

```

Kod źródłowy schematu komentarzy znajduje się na listingu 2.3

Listing 2.3: Schemat kolekcji komentarzy w postaci JSON dla bazy danych artykułów

```

1  {
2      $jsonSchema: {
3          bsonType: 'object',
4          title: 'comment',
5          required: [
6              'articleId',
7              'authorId',
8              'content',
9              'creationDate',
10             'modificationDate'
11         ],
12         properties: {
13             _id: {
14                 bsonType: 'objectId'
15             },
16             articleId: {
17                 bsonType: 'objectId'
18             },
19             parentCommentId: {
20                 bsonType: 'objectId'
21             },
22             authorId: {
23                 bsonType: 'int'
24             },
25             content: {
26                 bsonType: 'string'
27             },
28             creationDate: {
29                 bsonType: 'date'
30             },
31             modificationDate: {
32                 bsonType: 'date'
33             }
34         }
35     }
36 }

```

```
34     }
35   }
36 }
```

Na podstawie schematu komentarzy (listing 2.3) utworzono przykładowy dokument, który jest przedstawiony na listingu 2.4

Listing 2.4: Przykładowy dokument dla kolekcji komentarzy zgodny ze zdefiniowanym schematem

```
1  {
2    "_id": {
3      "$oid": "63618aa9973c760cacac05af"
4    },
5    "articleId": {
6      "$oid": "634308ab8f35533467caa4f7"
7    },
8    "parentCommentId": {
9      "$oid": "63617ca71676af3917d483fd"
10   },
11   "authorId": 27,
12   "content": "111",
13   "creationDate": {
14     "$date": {
15       "$numberLong": "1667336873772"
16     }
17   },
18   "modificationDate": {
19     "$date": {
20       "$numberLong": "1667336873772"
21     }
22   }
23 }
```

Kod źródłowy schematu opinii znajduje się na listingu 2.5

Listing 2.5: Schemat kolekcji opinii w postaci JSON dla bazy danych artykułów

```

1  {
2      $jsonSchema: {
3          bsonType: 'object',
4          title: 'opinion',
5          required: [
6              'author_id',
7              'rating',
8              'content',
9              'creationDate',
10             'modificationDate'
11         ],
12         properties: {
13             _id: {
14                 bsonType: 'objectId'
15             },
16             author_id: {
17                 bsonType: 'int'
18             },
19             rating: {
20                 bsonType: 'int',
21                 minimum: 1,
22                 maximum: 5
23             },
24             content: {
25                 bsonType: 'string'
26             },
27             creationDate: {
28                 bsonType: 'timestamp'
29             },
30             modificationDate: {
31                 bsonType: 'timestamp'
32             },
33             acceptances: {
34                 bsonType: 'array',
35                 items: {
36                     bsonType: 'object',
37                     required: [
38                         'author_id',
39                         'value'
40                     ],
41                     properties: {
42                         _id: {
43                             bsonType: 'objectId',
44                         },
45                         author_id: {
46                             bsonType: 'int'
47                         },
48                         value: {
49                             bsonType: 'int',
50                             'enum': [
51                                 -1,
52                                 1
53                             ]
54                         }
55                     }
56                 }
57             },
58             positiveAcceptancesCount: {
59                 bsonType: 'int'

```

```

60      },
61      negativeAcceptancesCount: {
62          bsonType: 'int'
63      }
64  }
65 }
66 }
```

Na podstawie schematu opinii (listing 2.5) utworzono przykładowy dokument, który jest przedstawiony na listingu 2.6

Listing 2.6: Przykładowy dokument dla kolekcji opinii zgodny ze zdefiniowanym schematem

```

1  {
2      "_id": {
3          "$oid": "6333266e3a8c25217485239c"
4      },
5      "author_id": 1,
6      "rating": 4,
7      "content": "Dobry artykuł, ale brakuje trochę o wersjach",
8      "creationDate": {
9          "$timestamp": {
10             "t": 323,
11             "i": 3128182693
12         }
13     },
14     "modificationDate": {
15         "$timestamp": {
16             "t": 323,
17             "i": 3128182693
18         }
19     },
20     "acceptances": [
21         {
22             "_id": {
23                 "$oid": "6333266e3a8c25217485239c"
24             },
25             "author_id": 2
26         },
27         {
28             "_id": {
29                 "$oid": "6333266e3a8c25217485239d"
30             },
31             "author_id": 3,
32             "value": -1
33         }
34     ]
35 }
```

# Rozdział 3

## Implementacja

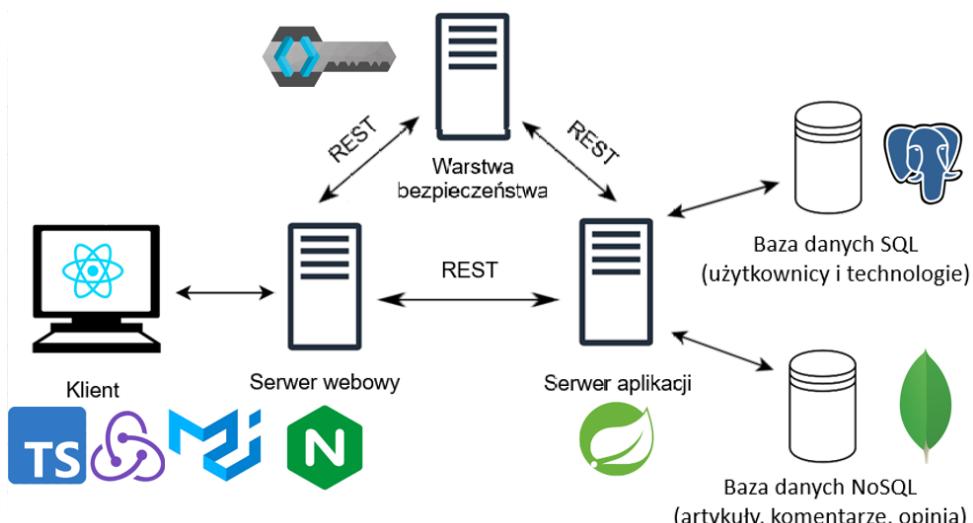
### 3.1. Technologie i narzędzia

W celu zaimplementowania zaprojektowanego systemu dokonano wyboru technologii i narzędzi. Przy ich wyborze kierowano się głównie ich odpowiedniością do realizowanego systemu, popularnością oraz własnym doświadczeniem uzyskanym w trakcie studiów. Z powodu tego, iż nie jest to projekt komercyjny, to poszukiwano darmowych rozwiązań. Innymi ważnymi czynnikami wyboru były również rozbudowana i czytelna dokumentacja, niezbyt duży poziom skomplikowania oraz bogata i aktywna społeczność wokół danego oprogramowania.

#### 3.1.1. Technologie

Użyto następujących technologii: PostgreSQL, MongoDB, Spring, Keycloak, NGINX, React, Typescript, Redux, MUI.

Wybrane technologie mogą być przyporządkowane do poszczególnych warstw systemu (Rys. 3.1).



Rys. 3.1: Użyte technologie

Źródło: opracowanie własne

## Baza danych

Jako system zarządzania obiektowo-relacyjną bazą danych wybrano PostgreSQL. PostgreSQL jest systemem darmowym, otwarto-źródłowym oraz niezwykle popularnym. PostgreSQL to najpopularniejszy RDBMS (ang. *Relational Database Management System*) tego typu, jak i jeden z najpopularniejszych otwartych RDBMS dostępnych na rynku. System ten ciągle się rozwija oraz jest niezawodny z powodu dosyć licznej społeczności zorientowanej wokół niego. Stanowi silną konkurencję dla innych baz danych pod względem bezpieczeństwa. Inną ważną zaletą tej technologii jest jej wysoka skalowalność.

Jako system zarządzania nierelacyjną bazą danych opartą na dokumentach wybrano MongoDB. MongoDB jest to otwarty i popularny system. MongoDB pozwala na wygodne przechowywanie danych w postaci klucz-wartość i nie wymaga użycia schematów, ale jest to zalecane. Dane są przechowywane jako dokumenty w postaci JSON. Dużymi zaletami tej technologii jest łatwość tworzenia danych oraz skalowalność. Z powodu wykorzystywanego formatu JSON, technologia ta jest wprost stworzona do użycia razem z językiem JavaScript, który jest wykorzystywany podczas realizacji tej pracy.

## Aplikacja serwerowa - Backend

Do realizacji aplikacji serwerowej wybrano język Java. Java jest wieloplatformowym, obiektowym i opartym na maszynie wirtualnej niezwykle popularnym językiem programowania. Język wykorzystuje się w bardzo wielu dziedzinach, jednak najczęściej są to aplikacje webowe oraz mobilne. Z powodu popularności tej technologii oraz jej długiego czasu na rynku (od 1996 roku), zebrała się dosyć duża społeczność wokół niej. Java posiada dużą liczbę bibliotek oraz frameworków, które przyspieszają pracę programisty.

Do realizacji aplikacji wykorzystano popularny Javowy framework Spring. Spring pozwala głównie na tworzenie backendu aplikacji webowych i jest obecnie bezkonkurencyjny w przypadku aplikacji tego typu. Pisanie aplikacji w Springu jest proste. Spring posiada bogatą i czytelną dokumentację wraz z przykładami użycia. Wokół Springa zebrała się duża społeczność. Inną zaletą tego framework jest to, że tak samo jak w przypadku Javy, dostępna jest duża liczba różnych bibliotek. Wadą tej technologii jest dosyć skomplikowana konfiguracja startowa aplikacji, lecz na szczęście istnieją technologie, które dostarczają tę konfigurację za programistę. Jedną z takich technologii jest framework Spring Boot, który dostarcza całą konfigurację aplikacji łącznie serwerem i od razu możliwe jest uruchomienie prostego backendu.

## Klient - Frontend

Do realizacji aplikacji klienckiej wybrano JavaScriptową bibliotekę React. React jest to popularna technologia pozwalająca tworzyć interfejsy graficzne aplikacji webowych. Głównym zastosowaniem tej technologii są aplikacje typu SPA. Tworzenie interfejsu graficznego w Reactcie jest proste i intuicyjne. Interfejs graficzny użytkownika tworzy się z tzw. komponentów. Komponenty pozwalają m.in. na ich ponowne użycie i dzięki temu można uniknąć niepotrzebnego powielania kodu.

JavaScript jest językiem słabo typowanym i przez to w trakcie komplikacji programu może być niewykryte wiele błędów. Mając na uwadze tę wadę, postanowiono wykorzystać technologię, która będzie umożliwiała wprowadzenie silnego typowania do aplikacji. Do tego celu wykorzystano Typescript, który jest nakładką na JavaScript. Typescript pozwala m.in. na tworzenie interfejsów, czy typowanie zmiennych.

W realizowanej aplikacji przewidziano system logowania oparty na tokenach na okaziciela i z tego powodu niezbędnym jest przechowywanie tych tokenów, aby możliwe było wysyłanie

żądań do zabezpieczonego backendu. Do tego celu idealnie pasuje Reactowa biblioteka Redux, która umożliwia przechowywanie odizolowanego od komponentów stanu aplikacji.

Do tworzenia ładnego interfejsu użytkownika wykorzystano popularną bibliotekę gotowych komponentów MUI. MUI zawiera bardzo dużą liczbę gotowych komponentów, których użycie jest proste. Biblioteka wspiera m.in. internacjonalizację, czy lokalizację.

### Serwer webowy

Do realizacji serwera webowego klienta wybrano serwer NGINX. NGINX jest to darmowy i popularny serwer webowy, proxy oraz IMAP/POP3. Serwer ten oparty jest na zdarzeniach (ang. *event-driven approach*), co oznacza, że może obsłużyć wiele żądań na jednym wątku. Serwer ten jest często wykorzystywany przez duże serwisy, jak np. Netflix, Instagram, czy GitHub.

### Warstwa bezpieczeństwa

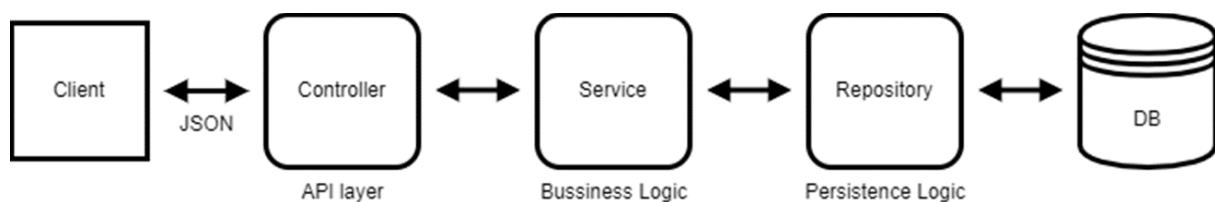
Do realizacji warstwy bezpieczeństwa wybrano technologię Keycloak. Keycloak jest to otwarty system, który implementuje uwierzytelnianie i autoryzacje. Keycloak wspiera logowanie SSO (ang. *single sign-on*), które umożliwia zalogowanie się do wielu serwisów za pomocą jednego konta. Keycloak jest najczęściej wykorzystywany w formie osobnego serwera.

### 3.1.2. Narzędzia

Użyto następujących narzędzi:

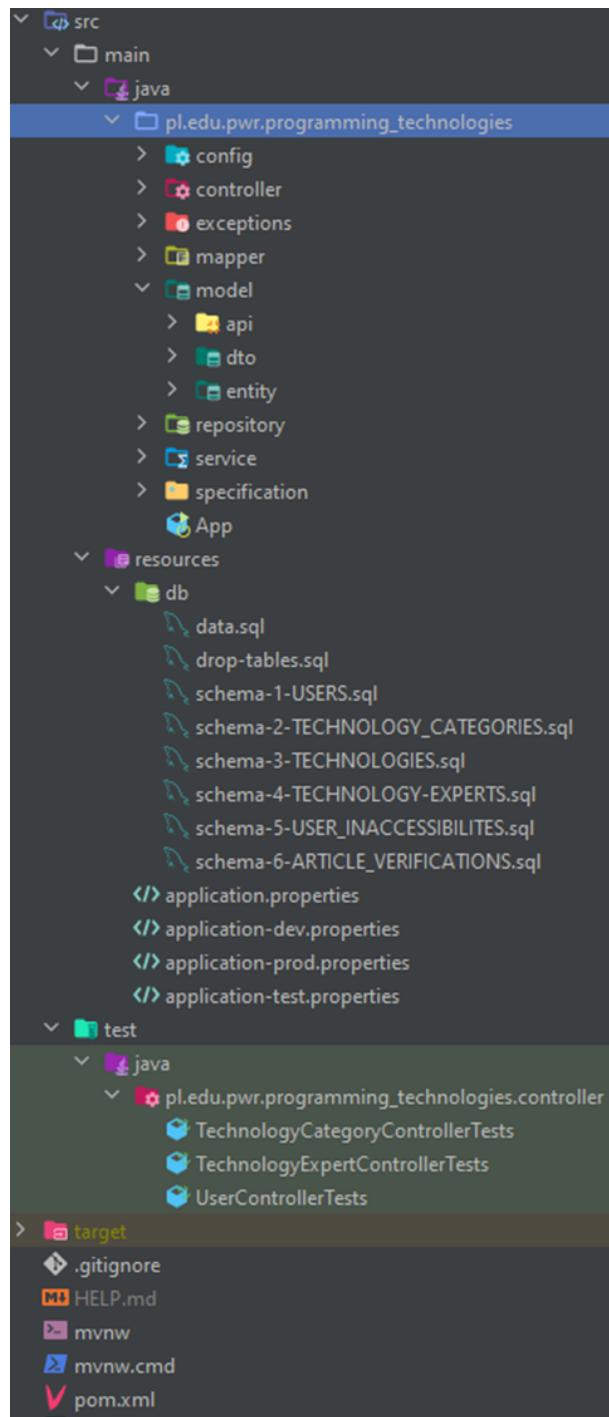
1. LaTeX - napisanie dokumentacji,
2. Visual Paradigm - sporządzenie diagramów,
3. Figma - zaprojektowanie interfejsu użytkownika,
4. pgAdmin - zarządzanie bazą PostgreSQL,
5. MongoDBCompass - zarządzanie bazą MongoDB,
6. Visual Studio Code - środowisko programistyczne dla frontendu,
7. ItelliJ - środowisko programistyczne dla backendu,
8. REST-assured - testy integracyjne backendu,
9. Selenium Web Driver - testy automatyczne frontendu,
10. Docker - stworzenie i umieszczenie kontenerów aplikacji w rejestrze kontenerów,
11. MongoDB Cloud - hostowanie bazy MongoDB,
12. Azure - hostowanie całego systemu oprócz bazy MongoDB.

## 3.2. Aplikacja serwerowa



Rys. 3.2: Architektura aplikacji serwerowej

Źródło: opracowanie własne



Rys. 3.3: Struktura projektu aplikacji serwerowej

Źródło: opracowanie własne

```

17     @Builder
18     @Data
19     @NoArgsConstructor
20     @AllArgsConstructor
21     @Document("comments")
22     public class CommentEntity {
23
24         @Id
25         @MongoId(FieldType.OBJECT_ID)
26         @Field(name = "_id")
27         private ObjectId id;
28
29         @NotNull(message = "Nie podano id artykułu")
30         @Field(name = "articleId")
31         private ObjectId articleId;
32
33         @Field(name = "parentCommentId")
34         private ObjectId parentCommentId;
35
36         @NotNull(message = "Nie podano id autora")
37         @Field(name = "authorId")
38         private Integer authorId;
39
40         @NotNull(message = "Nie podano zawartości komentarza")
41         @Field(name = "content")
42         private String content;
43
44         @NotNull
45         @Field(name = "creationDate")
46         private LocalDateTime creationDate;
47
48         @NotNull
49         @Field(name = "modificationDate")
50         private LocalDateTime modificationDate;
51     }

```

Rys. 3.4: Model komentarza w aplikacji serwerowej

Źródło: opracowanie własne

```

25     @Slf4j
26     @Service
27     @RequiredArgsConstructor
28     public class CommentServiceImpl implements CommentService {
29
30         private final CommentRepository commentRepository;
31         private final ArticleRepository articleRepository;
32         private final UserRepository userRepository;
33
34         @Override
35         public Page<CommentEntity> getSubComments(ObjectId articleId, ObjectId parentCommentId, Pageable pageable) {
36             return commentRepository.findAllByArticleIdAndParentCommentId(articleId, parentCommentId, pageable);
37         }

```

Rys. 3.5: Serwis dla komentarzy w aplikacji serwerowej

Źródło: opracowanie własne

```

21  @RestController
22  @CrossOrigin(origins = {"http://localhost:3000", "https://technologie-programistyczne.netlify.app"})
23  @RequiredArgsConstructor
24  @RequestMapping(value = "/comment")
25  public class CommentController {
26
27      private final CommentService commentService;
28      private final CommentMapper commentMapper = CommentMapper.INSTANCE;
29      private final UserRepository userRepository;
30
31      @GetMapping("/subs/{parentCommentId}")
32      public ResponseEntity getSubComments(
33          @PathVariable("parentCommentId") String parentCommentIdStr, Pageable pageable
34      ){
35          ObjectId parentCommentId;
36
37          try{
38              parentCommentId = new ObjectId(parentCommentIdStr);
39          }
40          catch(IllegalArgumentException e){
41              return ResponseEntity.status(HttpStatus.BAD_REQUEST).body("Podano niewłaściwe id komentarza");
42          }
43
44          if(pageable == null){
45              pageable = Pageable.unpaged();
46          }
47
48          Page<CommentEntity> foundCommentEntityPage = commentService.getSubComments(parentCommentId, pageable);
49          Page<CommentDTO> foundCommentDTOPage = foundCommentEntityPage.map(c ->
50              commentMapper.commentEntityToCommentDTO(c, userRepository)
51          );
52
53
54      }
}

```

Rys. 3.6: Kontroler dla komentarzy w aplikacji serwerowej

Źródło: opracowanie własne

```

13  @Mapper(uses = {DateTimeMapper.class, UserMapper.class, MongoObjectIDMapper.class})
14  public interface CommentMapper {
15
16      CommentMapper INSTANCE = Mappers.getMapper(CommentMapper.class);
17
18      @Mapping(source = "id", target = "id", qualifiedByName = "objectIdToString")
19      @Mapping(source = "articleId", target = "articleId", qualifiedByName = "objectIdToString")
20      @Mapping(source = "parentCommentId", target = "parentCommentId", qualifiedByName = "objectIdToString")
21      @Mapping(
22          source = "creationDate",
23          target = "creationDate",
24          qualifiedByName = "localDateTimeToOffsetDateTime"
25      )
26      @Mapping(
27          source = "modificationDate",
28          target = "modificationDate",
29          qualifiedByName = "localDateTimeToOffsetDateTime"
30      )
31      @Mapping(source = "authorId", target = "authorDTO", qualifiedByName = "userIdToUserDTO")
32      CommentDTO commentEntityToCommentDTO(CommentEntity commentEntity, @Context UserRepository userRepository);
33
34

```

Rys. 3.7: Mapowanie między modelem komentarza, a dto w aplikacji serwerowej

Źródło: opracowanie własne

user-controller	
PUT	/user/{userId}
POST	/user
GET	/user/user-account-id/{userAccountId}
GET	/user/nickname/{nickname}
comment-controller	
PUT	/comment/{commentId}
DELETE	/comment/{commentId}
POST	/comment
GET	/comment/subs/{parentCommentId}
article-controller	
GET	/article/{articleId}
PUT	/article/{articleId}
DELETE	/article/{articleId}
GET	/article
POST	/article
POST	/article/search
GET	/article/{articleId}/comments/parents

Rys. 3.8: Endpointy dla użytkowników, komentarzy oraz artykułów w aplikacji serwerowej

Źródło: opracowanie własne

article-verification-controller	
GET	/article-verification/{articleVerificationId}
PUT	/article-verification/{articleVerificationId}
GET	/article-verification/to/{reviewerId}
user-inaccessibility-controller	
POST	/user-inaccessibility
GET	/user-inaccessibility/user/{userId}
DELETE	/user-inaccessibility/{userInaccessibilityId}
technology-expert-controller	
GET	/technology-expert/user/{userId}
POST	/technology-expert/user/{userId}
GET	/technology-expert/user/{userId}/assignable-technologies
GET	/technology-expert/user/{userId}/assignable-technologies/technology-category/{technologyCategoryId}
DELETE	/technology-expert/{technologyExpertId}

Rys. 3.9: Endpointy dla weryfikacji artykułów, dostępności użytkowników oraz technologii, w których specjalizują się recenzenci w aplikacji serwerowej

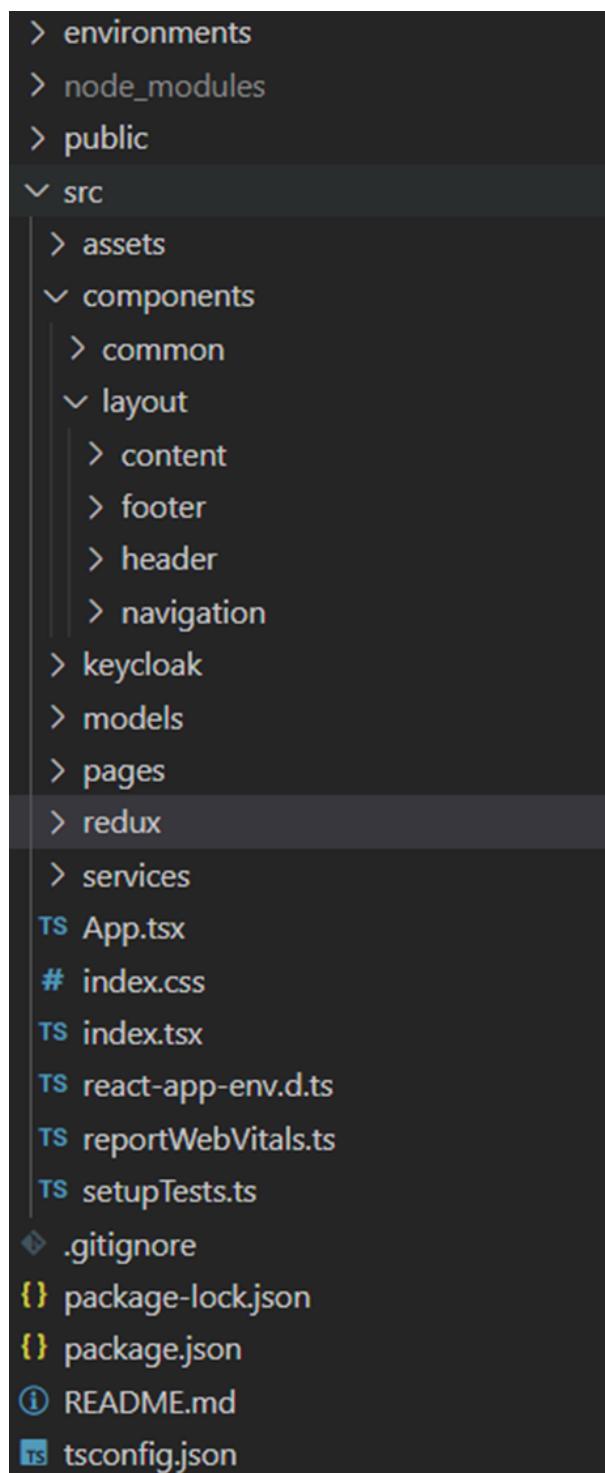
Źródło: opracowanie własne

technology-controller	
GET	/technology
GET	/technology/category/{technologyCategoryId}
technology-category-controller	
GET	/technology-category/{technologyCategoryId}
GET	/technology-category/tree

Rys. 3.10: Endpointy dla technologii i kategorii technologii w aplikacji serwerowej

Źródło: opracowanie własne

### 3.3. Aplikacja kliencka



Rys. 3.11: Struktura projektu aplikacji klienckiej

Źródło: opracowanie własne

```

3  export interface Technology {
4      id: number,
5      name: string,
6      description: string,
7      creationDateTime: Date,
8      modificationDate: Date,
9      provider?: string,
10     icon?: string,
11     firstReleaseDateTime: Date,
12     lastReleaseDateTime: Date,
13     technologyCategoryDTO: TechnologyCategory
14 }

```

Rys. 3.12: Model technologii w aplikacji klienckiej

Źródło: opracowanie własne

```

5  export interface UserState {
6      user: any,
7  }
8
9  const initialState: UserState = {
10    user: {},
11 }
12
13 export const userSlice = createSlice({
14     name: 'users',
15     initialState,
16     reducers: {
17         setUser(state, action: PayloadAction<User | {}>){
18             state.user = action.payload
19         }
20     }
21 })
22
23 export const { setUser } = userSlice.actions
24
25 export default userSlice.reducer

```

Rys. 3.13: Slice zalogowanego użytkownika w aplikacji klienckiej

Źródło: opracowanie własne

```

8  const NavBar = () => {
9
10    const actualRoles = useSelector((state: RootState) => state.keycloak.roles)
11
12    return [
13      <nav>
14        <Toolbar>
15          <Typography variant="h6" align="center" component="div" sx={{ flexGrow: 1, ml: 2 }}>
16            <Link to="/articles">
17              Artykuły
18            </Link>
19          </Typography>
20          <Typography variant="h6" align="center" component="div" sx={{ flexGrow: 1, ml: 2 }}>
21            <Link to="/technologies">
22              Technologie
23            </Link>
24          </Typography>
25          {actualRoles.includes(roles.reviewer.name) ?
26            <Typography variant="h6" align="center" component="div" sx={{ flexGrow: 1, ml: 2 }}>
27              <Link to="/articles-verification">
28                Weryfikacja artykułów
29              </Link>
30            </Typography>
31          : null
32        }

```

Rys. 3.14: Komponent nawigacja w aplikacji klienckiej

Źródło: opracowanie własne

```

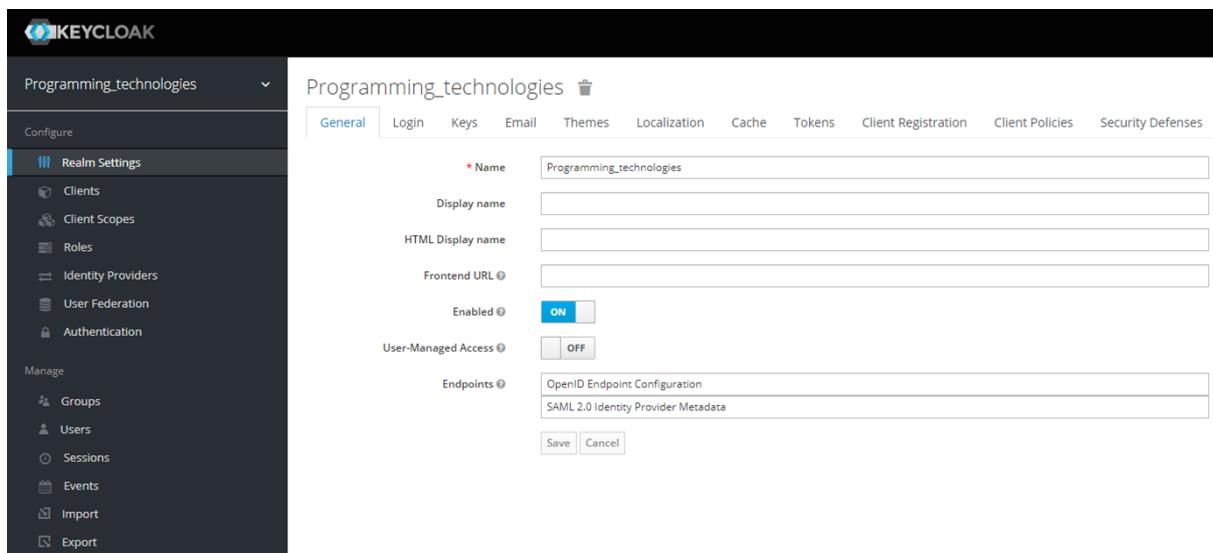
1, import axios from "axios";
2
3 class TechnologyAPIService {
4
5   private apiUrl: string = `${process.env.REACT_APP_API as string}/technology`
6
7   getAll = () => {
8     return axios.get(this.apiUrl)
9   }
10
11  getAllByTechnologyCategoryId = (technologyCategoryId: number) => {
12    return axios.get(`${this.apiUrl}/category/${technologyCategoryId}`)
13  }
14
15}
16
17 export default new TechnologyAPIService();

```

Rys. 3.15: Serwis API dla technologii w aplikacji klienckiej

Źródło: opracowanie własne

## 3.4. Warstwa bezpieczeństwa



Rys. 3.16: Panel administratora Keycloka

Źródło: opracowanie własne

## **Rozdział 4**

# **Testowanie**

## **Rozdział 5**

# **Podsumowanie**

# **Literatura**

## **Dodatek A**

### **Instrukcja wdrożeniowa**

## **Dodatek B**

### **Opis załączonej płyty CD/DVD**