

Politechnika Wrocławska
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

1. Wstęp	8
1.1. Wprowadzenie	8
1.2. Cel pracy	11
1.3. Zakres pracy	11
2. Projektowanie	12
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	76
3. Implementacja	82
3.1. Technologie i narzędzia	82
3.1.1. Technologie	82
3.1.2. Narzędzia	84
3.2. Aplikacja serwerowa	84
4. Testowanie	92
5. Wdrożenie	93
6. Podsumowanie	94
Literatura	95
A. Instrukcja wdrożeniowa	96
B. Opis załączonej płyty CD/DVD	97

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 artykule	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 ustawiona 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 albo 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	55
2.40. Panel z zawartością artykułu widziany z perspektywy administratora	58
2.41. Panel z listą komentarzy o artykule	59
2.42. Formularz dodania albo edytowania komentarza o artykule	60
2.43. Dostępne opcje 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. Formularz dodania albo 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 do wyszukiwania technologii z podanymi kryteriami wyszukiwania oraz listą wyszukanych technologii widziany z perspektywy recenzenta	73
2.52. Panel z formularzem dodania albo modyfikacji technologii	74
2.53. Diagram ERD dla głównej bazy danych	76
3.1. Użyte technologie	82
3.2. Architektura aplikacji serwerowej	85
3.3. Struktura projektu aplikacji serwerowej	86

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

Spis listingów

2.1.	Schemat kolekcji artykułów w postaci JSON dla bazy danych artykułów	77
2.2.	Przykładowy dokument dla kolekcji artykułów zgodny ze zdefiniowanym schematem	78
2.3.	Schemat kolekcji komentarzy w postaci JSON dla bazy danych artykułów	78
2.4.	Przykładowy dokument dla kolekcji komentarzy zgodny ze zdefiniowanym schematem	79
2.5.	Schemat kolekcji opinii w postaci JSON dla bazy danych artykułów	80
2.6.	Przykładowy dokument dla kolekcji opinii zgodny ze zdefiniowanym schematem .	81
3.1.	Profil lokalny	86
3.2.	Encja technologii	87
3.3.	Mapowanie jeden do wielu między technologiami i kategoriami technologii	88
3.4.	Encja artykułu	88
3.5.	Repozytorium technologii	89
3.6.	Repozytorium technologii	89
3.7.	Repozytorium technologii	90
3.8.	Repozytorium technologii	90

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.



Rys. 1.1: Wikipedia

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

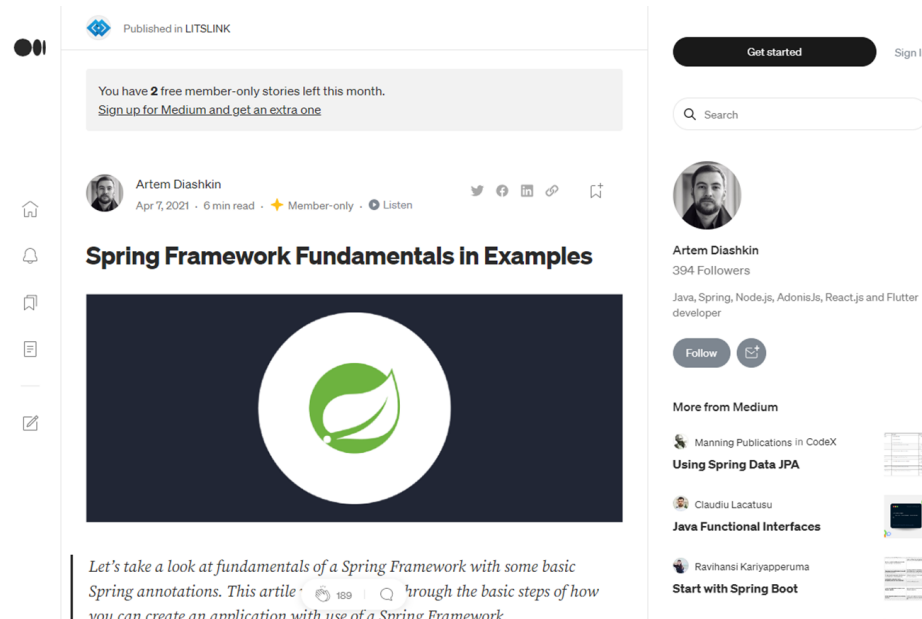
Można wyróżnić następujące zalety Wikipedii:

- 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 górę i w dół albo sama pochwała), jak i często nie ma wymogu zamieszczenia tekstowego 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 нефункциональных, 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 нефункциональных. W tym celu konieczne jest wprowadzenie pojęć, które będą wykorzystywane w dalszej części pracy.

2.1.1. Słownik pojęć

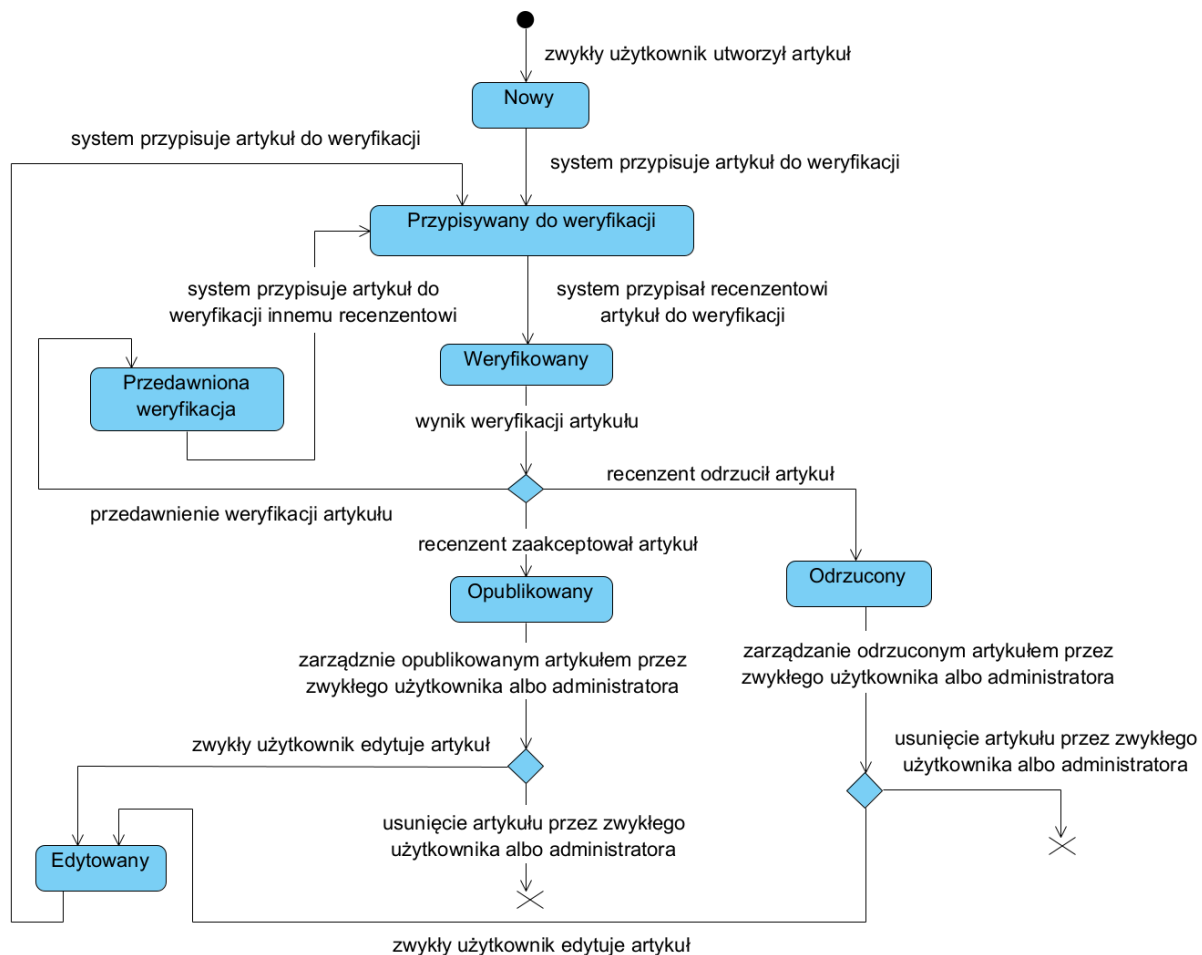
Przed zdefiniowaniem wymagań funkcjonalnych, potrzebne było określenie pojęć, które będą używane zarówno przy wymaganiach funkcjonalnych, jak i w reszcie pracy.

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, przedawniona 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 górę) oznaczająca, że użytkownik zgadza się z opinią i ocena negatywna (łapka w dół) oznaczająca, 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,
- **Niezałogowany użytkownik** - niezałogowany 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 przeprowadzanej 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) Awatar,
 - 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 akceptację 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

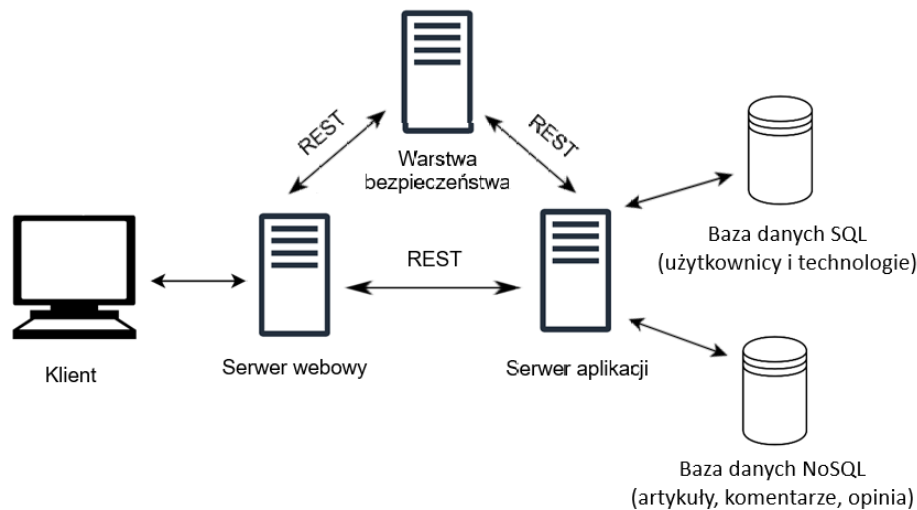
Na podstawie celu oraz zakresu pracy zdefiniowano następujące wymagania niefunkcjonalne:

- 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 dlatego, że taka forma aplikacji zapewnia dużą wygodę użytkownika 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 poświadczenia 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. W realizowanym systemie wykryto wiele przypadków, w których można utworzyć relacje np. 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 wykorzystywanymi narzędziami (np. framework Hibernate języka Java), które są oparte na programowaniu obiektowym.

Artykuły mają złożoną zagnieżdżoną strukturę, zawierają 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 zagnieżdżonych struktur oraz umożliwia pobranie danych w postaci zagnieżdżonej 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 przesłać żądanie. Punkty końcowe posiadają adresy URL np. `http://localhost:9000/user`. Adresy URL składają się z 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

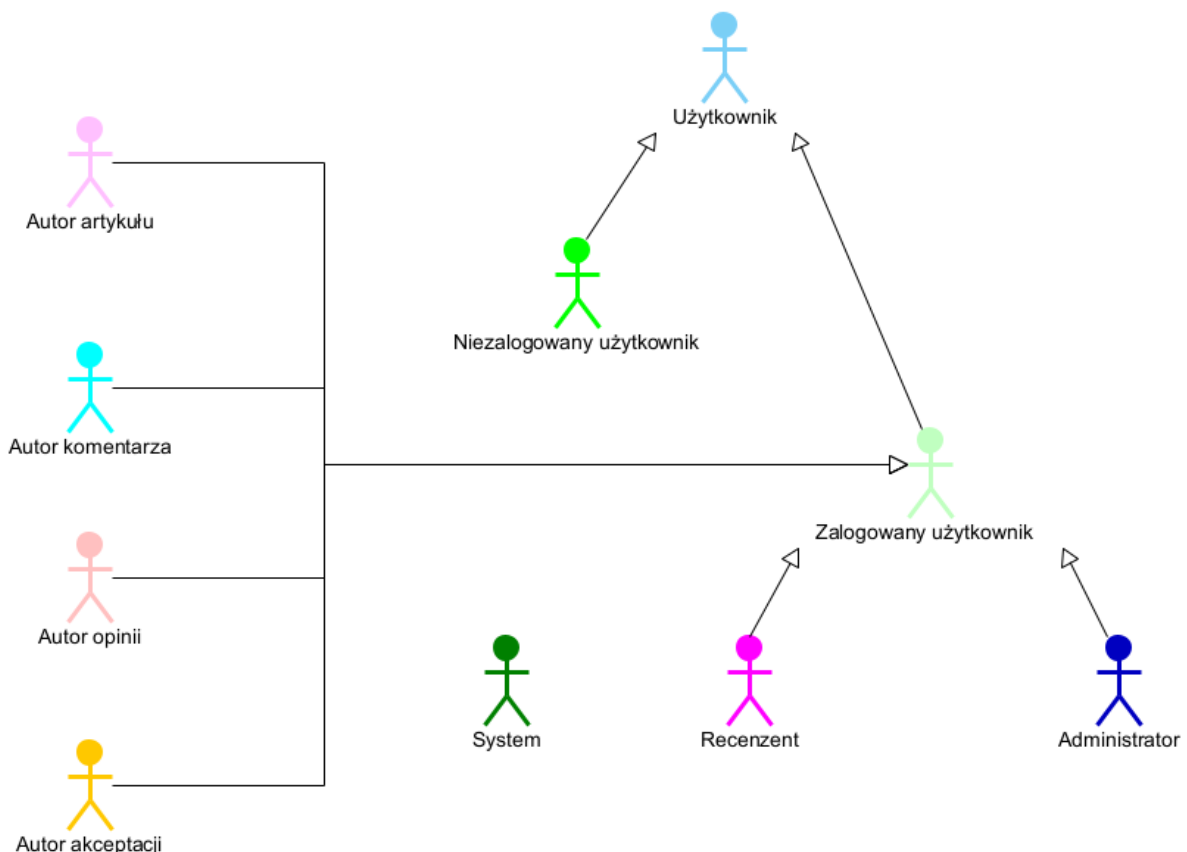
Na podstawie analizy wymagań funkcjonalnych zdefiniowano następujących aktorów:

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

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

2.4.2. Diagramy przypadków użycia

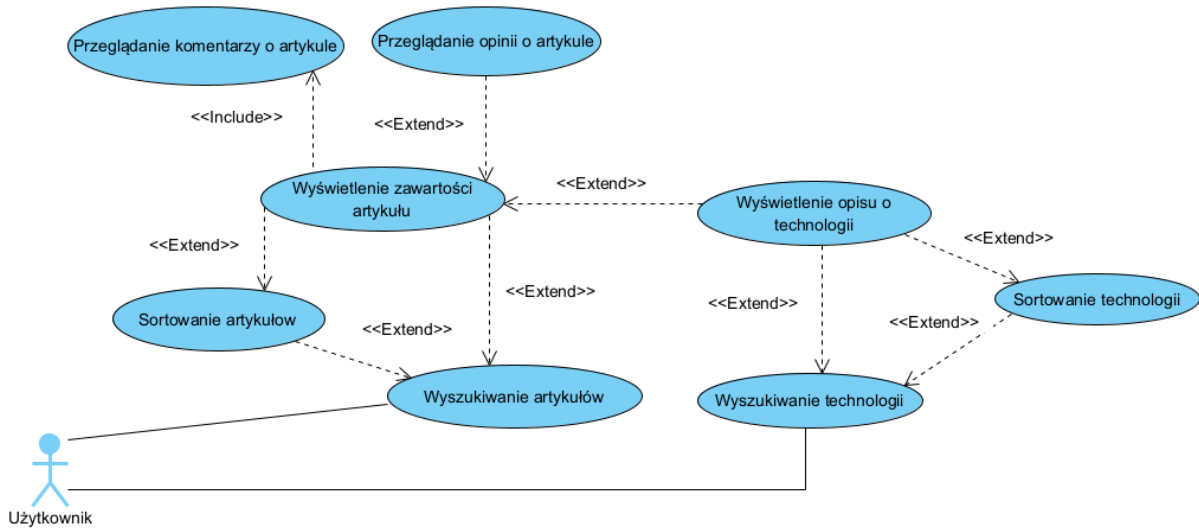
Na podstawie zdefiniowanych wcześniej wymagań funkcjonalnych oraz zidentyfikowanych aktorów, sporządzono diagramy przypadków użycia. Zdecydowano się na zastosowanie dziedziczenia między aktorami, gdyż opracowane role charakteryzują się tym, że posiadają wiele wspólnych uprawnień (przypadków użycia) i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



Rys. 2.3: Dziedziczenie między aktorami

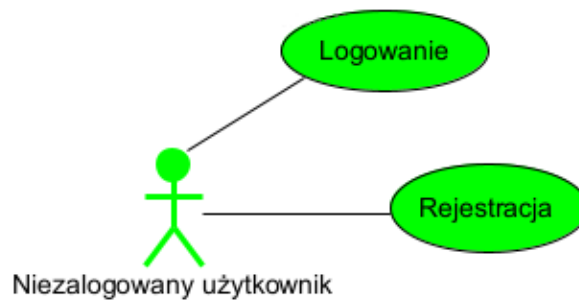
Źródło: opracowanie własne

Z powodu 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.



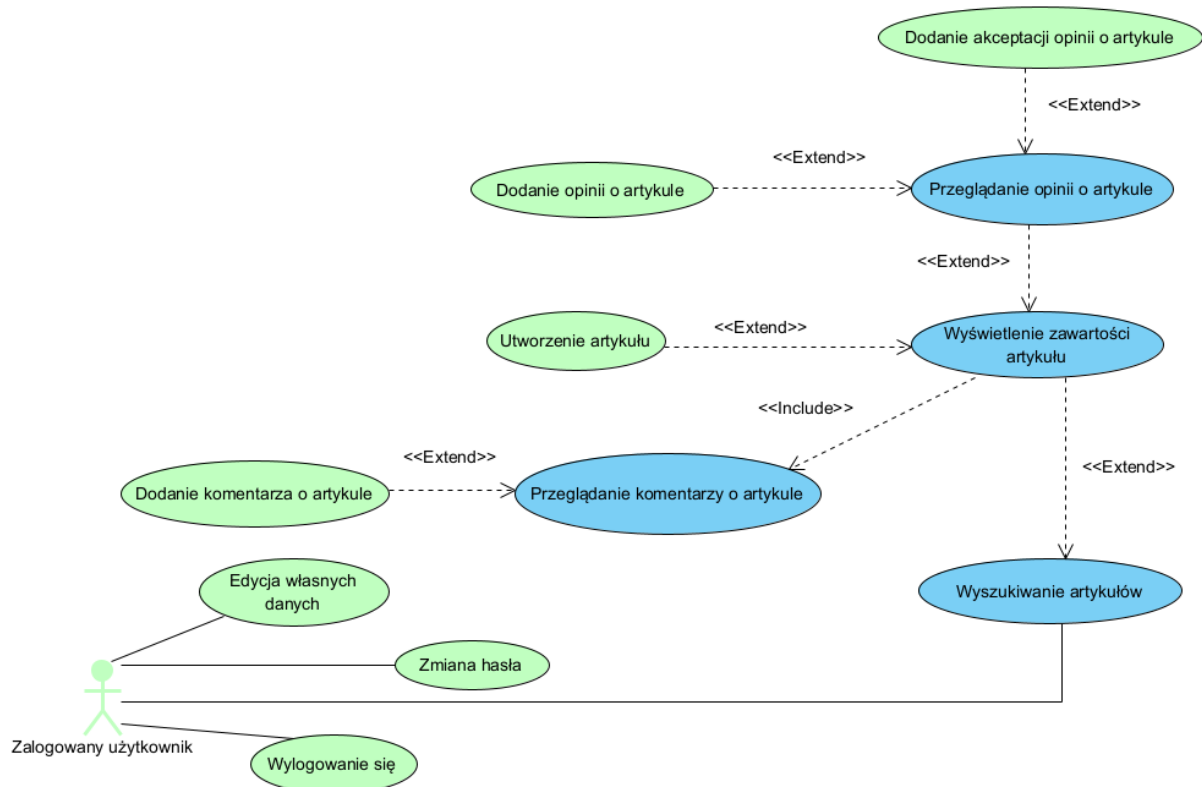
Rys. 2.4: Diagram przypadków użycia dla każdego użytkownika

Źródło: opracowanie własne



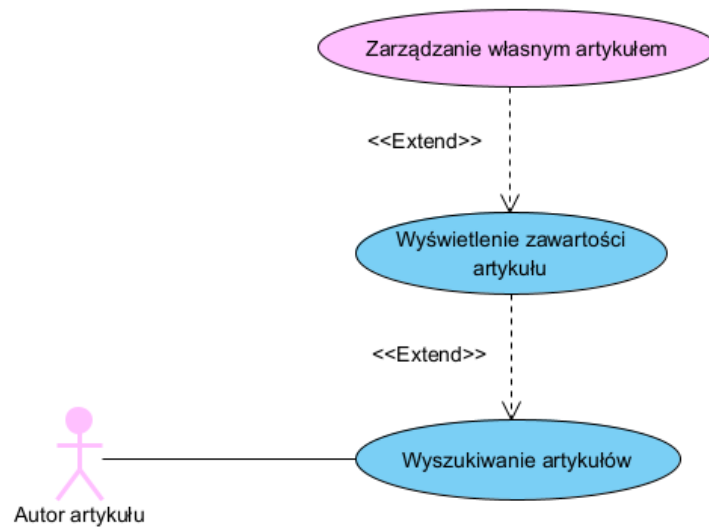
Rys. 2.5: Diagram przypadków użycia dla niezalogowanego użytkownika

Źródło: opracowanie własne



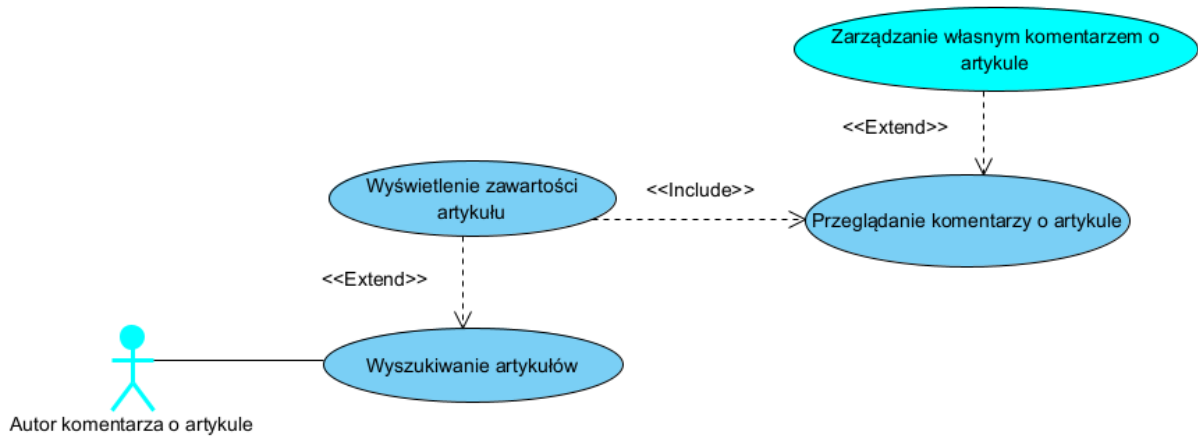
Rys. 2.6: Diagram przypadków użycia dla zalogowanego użytkownika

Źródło: opracowanie własne



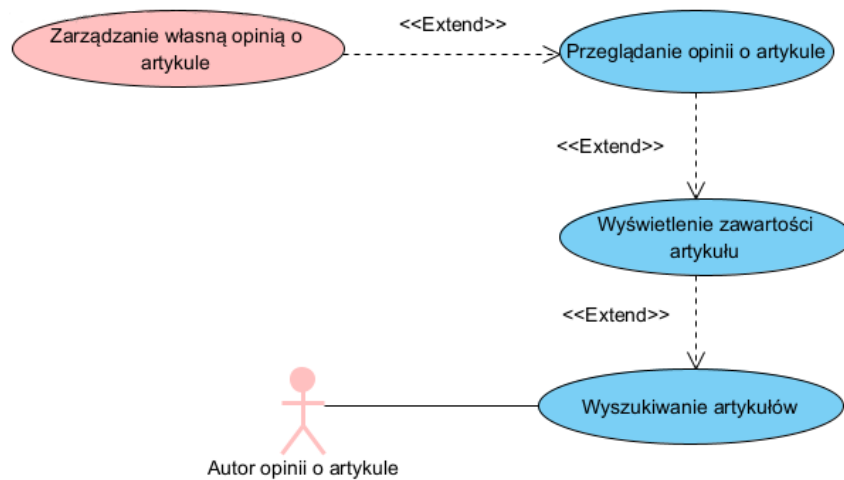
Rys. 2.7: Diagram przypadków użycia dla autora artykułu

Źródło: opracowanie własne



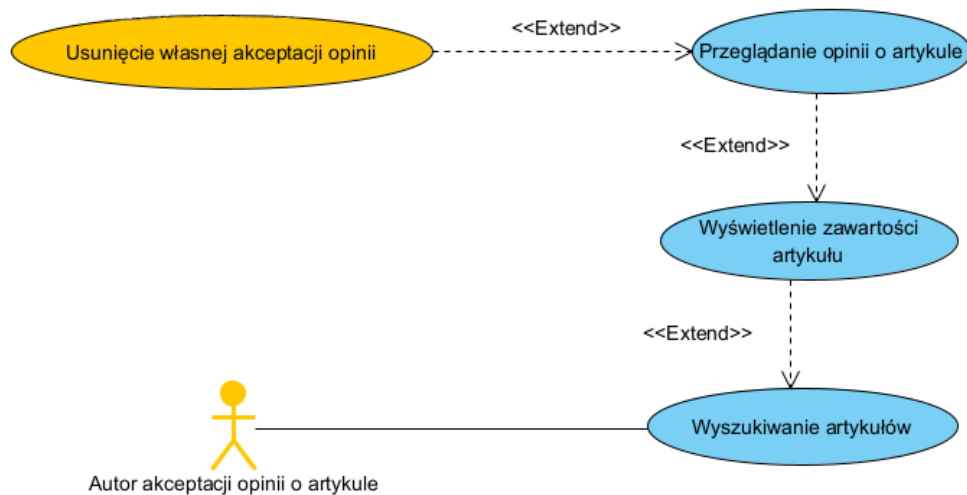
Rys. 2.8: Diagram przypadków użycia dla autora komentarza o artykule

Źródło: opracowanie własne



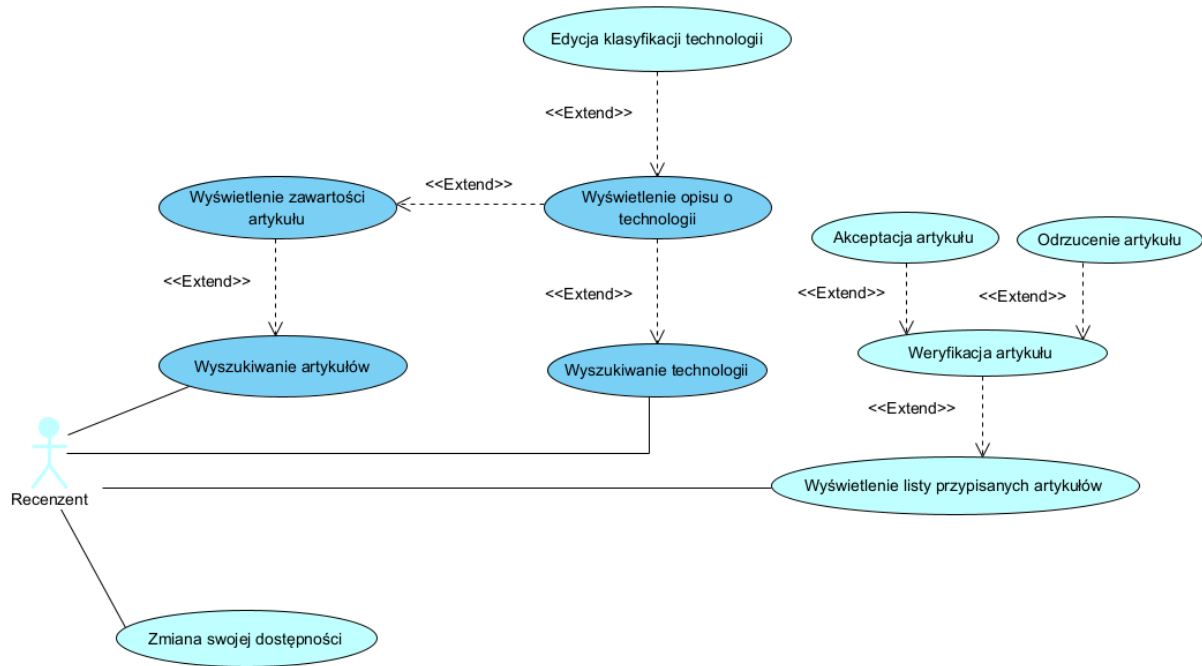
Rys. 2.9: Diagram przypadków użycia dla autora opinii o artykule

Źródło: opracowanie własne



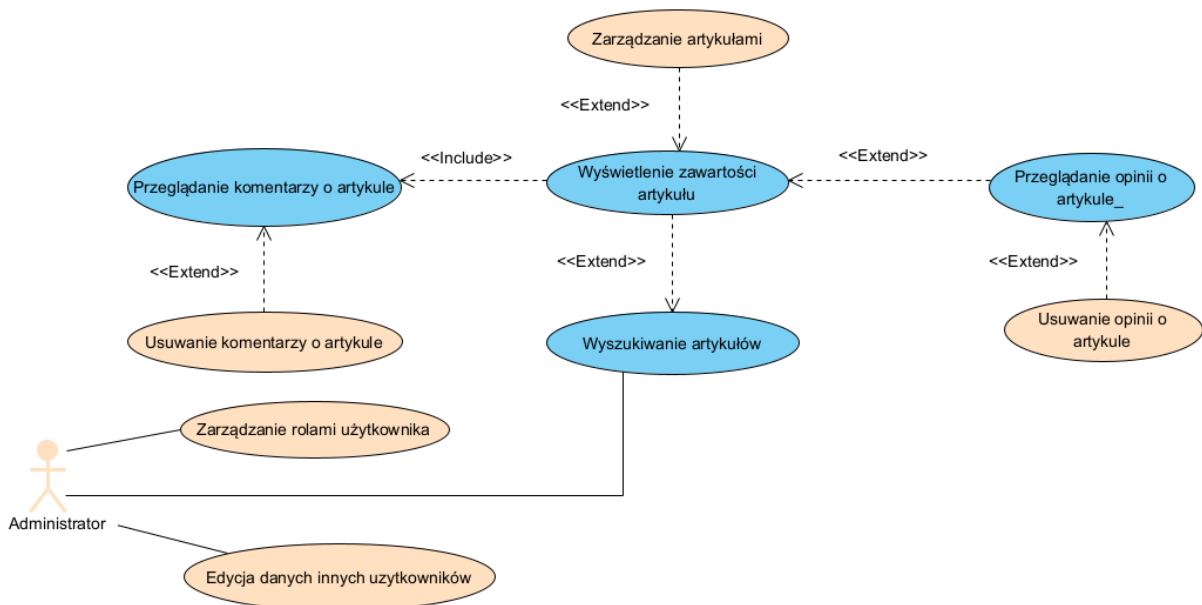
Rys. 2.10: Diagram przypadków użycia dla autora akceptacji opinii artykułu

Źródło: opracowanie własne



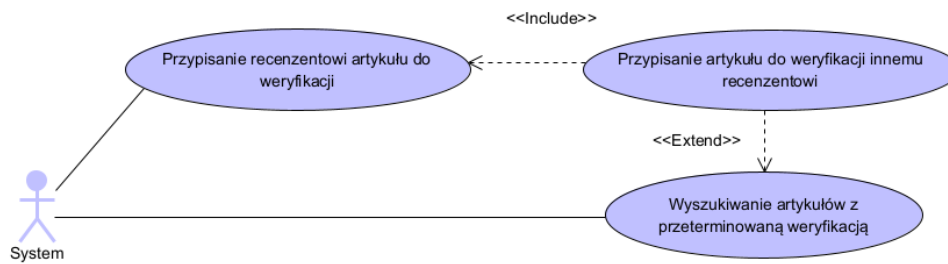
Rys. 2.11: Diagram przypadków użycia dla recenzenta

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:	Niezałogowany użytkownik
Warunki początkowe:	Użytkownik jest niezałogowany
Warunki końcowe:	Założenie konta zwykłego użytkownika
Przebieg:	<ol style="list-style-type: none"> 1. Użytkownik klika przycisk „Rejestracja” na nagłówku strony (Rys. 2.14), 2. Aplikacja wyświetla formularz do rejestracji (Rys. 2.15), 3. Użytkownik wypełnia pola w formularzu, 4. Użytkownik klika przycisk „Zarejestruj”, 5. Serwis przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych, 6. Po pozytywnej walidacji danych, aplikacja wyświetla regulamin serwisu, 7. Użytkownik przegląda cały regulamin i go akceptuje poprzez kliknięcie w pole wyboru „Akceptuje regulamin” (Rys. 2.16), 8. Serwis tworzy nowe konto.

Alternatywny przebieg:	<p>Użytkownik anuluje rejestrację</p> <p>3a. Użytkownik klika przycisk „X” albo klika 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”, klika przycisk „X” albo klika w obszar poza formularzem,</p> <p>8c. Wyłączenie formularza.</p>

IT Tech Technologie IT <div> Logowanie Rejestracja </div>	
Artykuły	Technologie

Rys. 2.14: Nagłówek strony niezalogowanego użytkownika

Źródło: opracowanie własne

X

Rejestracja

Imię

*

Nazwisko

*

Nazwa użytkownika

*

Pseudonim

*

E-mail

*

Hasło

*

Powtórz hasło

*

K

Dodaj awatar

+

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

Rys. 2.17: Formularz logowania

Źródło: opracowanie własne

Rys. 2.18: Nagłówek strony zalogowanego użytkownika

Źródło: opracowanie własne

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"> 1. Użytkownik klika na swój awatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta (Rys. 2.20), 2. Użytkownik klika przycisk „Wyloguj się”, 3. Aplikacja wyświetla komunikat o wylogowaniu się, 4. Następuje wylogowanie użytkownika oraz zmienia się wygląd nagłówka strony (Rys. 2.14).
-----------	---



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"> 1. Użytkownik klika na swój awatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta (Rys. 2.20), 2. Użytkownik klika przycisk „Dane użytkownika”, 3. Aplikacja wyświetla panel z danymi użytkownika (Rys 2.21), 4. Użytkownik klika przycisk „Edytuj dane”, 5. Serwis wyświetla formularz edycji danych (Rys 2.22), 6. Użytkownik zmienia dane w formularzu, 7. Użytkownik klika przycisk „Zapisz”, 8. Serwis przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych, 9. Po pozytywnej walidacji danych, aplikacja zapisuje zmienione dane użytkownika.


Alternatywny przebieg:	<p>Użytkownik anuluje edycję danych</p> <p>6a. Użytkownik klika przycisk „Anuluj”,</p> <p>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,</p> <p>10b. Powrót do punktu 6.</p>

IT Tech Technologie IT
 kamil_dywan

Artykuły

Technologie

Dane użytkownika

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

Edytuj dane

Rys. 2.21: Panel z danymi użytkownika

Źródło: opracowanie własne

IT Tech

Technologie IT

kamil_dywan

Artykuły

Technologie

Dane użytkownika

Imię

Kamil

Nazwisko

Dywan

Nazwa użytkownika

kamil.dywan

Pseudonim

kamil_dywan

E-mail

kamil.dywan@mail.com

Awatar

K

avatar.png +

Zapisz

Anuluj

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"> 1. Użytkownik klika na swój awatar dostępny na nagłówku strony aby wyświetlić opcje dotyczące konta (Rys. 2.20), 2. Użytkownik klika przycisk „Zmień hasło”, 3. Aplikacja wyświetla formularz do zmiany hasła (Rys. 2.23) 4. Użytkownik podaje aktualne oraz nowe hasło, 5. Użytkownik klika przycisk „Zapisz”, 6. 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, 7. Po pozytywnej walidacji hasło użytkownika zostaje zmienione.
Alternatywny przebieg:	<p>Użytkownik anuluje zmianę hasła</p> <ol style="list-style-type: none"> 4a. Użytkownik klika 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 tytułem "Zmiana hasła". Zawiera dwa pola tekstowe: "Aktualne hasło" i "Nowe hasło". Poniżej znajdują się dwa przyciski: "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"> 1. Użytkownik klika przycisk „Zarządzanie użytkownikami” znajdujący się w menu głównym strony (Rys. 2.24), 2. Aplikacja wyświetla podstronę z wyszukiwarką użytkowników, 3. Użytkownik wpisuje nazwę użytkownika (np. kamil), 4. Użytkownik klika przycisk „Szukaj”, 5. Serwis wyszukuje użytkowników, 6. Aplikacja wyświetla listę wyszukanych użytkowników (Rys 2.25) 7. Użytkownik klika na jednego z innych użytkowników, 8. Aplikacja wyświetla panel z danymi wybranego użytkownika (Rys 2.26), 9. Użytkownik klika przycisk „Edytuj dane”, 10. Serwis wyświetla formularz edycji danych (Rys 2.27), 11. Użytkownik zmienia dane w formularzu, 12. Użytkownik klika przycisk „Zapisz”, 13. Serwis przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych, 14. Po pozytywnej walidacji danych, aplikacja zapisuje zmienione dane innego użytkownika.
Alternatywny przebieg:	<p>Użytkownik resetuje hasło innego użytkownika</p> <ol style="list-style-type: none"> 9a. Użytkownik klika przycisk „Reset hasła”, 10a. Aplikacja zastępuje hasło wybranego użytkownika pustym ciągiem znaków, 11a. Powrót do punktu 8.
Alternatywny przebieg:	<p>Użytkownik anuluje edycję danych</p> <ol style="list-style-type: none"> 11b. Użytkownik klika przycisk „Anuluj”, 12b. Aplikacja wyświetla panel z danymi użytkownika (Rys. 2.26).
Alternatywny przebieg:	<p>Wprowadzone przez użytkownika dane nie przeszły walidacji</p> <ol style="list-style-type: none"> 14c. Aplikacja zmienia kolor obramowania pól, które nie przeszły walidacji, na czerwono oraz wyświetla pod tymi polami adekwatny komunikat o błędzie, 15c. Powrót do punktu 11.

IT Tech Technologie IT		Kamil Dywan 
Artykuły	Technologie	Zarządzanie użytkownikami

Rys. 2.24: Nagłówek strony administratora

Źródło: opracowanie własne

IT Tech Technologie IT		Kamil Dywan 
Artykuły	Technologie	Zarządzanie użytkownikami

Wyszukiwanie użytkowników

Nazwa użytkownika

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

Rys. 2.25: Panel do wyszukiwania użytkowników

Źródło: opracowanie własne

IT Tech

Technologie IT

Kamil Dywan

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

Dane użytkownika

Imię	Kamil	
Nazwisko	Dywan	
Nazwa użytkownika	kamil.dywan	Awatar
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

IT Tech

Technologie IT

Kamil Dywan

Artykuły

Technologie

Zarządzanie użytkownikami

Dane użytkownika

Imię

Kamil

Nazwisko

Dywan

Nazwa użytkownika

kamil.dywan

Pseudonim

kamil_dywan

E-mail

kamil.dywan@mail.com

Awatar

K

avatar.png +

Reset hasła

Zapisz

Anuluj

Role użytkownika

Recenzent

☒

Administrator

☐

Rys. 2.27: Panel umożliwiający administratorowi edycję danych użytkownika

Źródło: opracowanie własne

Tab. 2.7: Opis przypadku użycia - zarządzanie rolami użytkownika

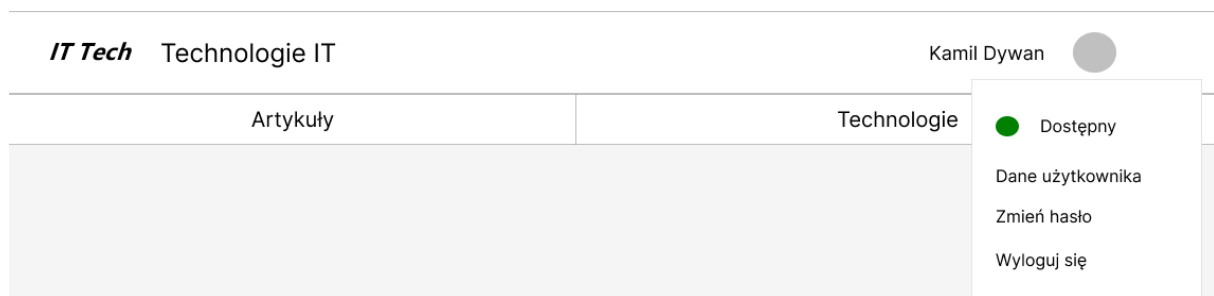
Nazwa:	Zarządzanie rolami innego użytkownika
Cel:	Zmiana przypisanych ról innego użytkownika
Aktorzy:	Administrator
Warunki początkowe:	Wyświetlony panel z danymi innego użytkownika (Rys 2.26)
Warunki końcowe:	Zmiana przypisanych ról innego użytkownika

Przebieg:	<p>Inny użytkownik posiada wybraną rolę (np. Recenent na Rys 2.26)</p> <ol style="list-style-type: none"> 1. Użytkownik klika 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), 2. Pole wyboru staje się puste, 3. Aplikacja usuwa przypisanie roli innego użytkownika.
Alternatywny przebieg:	<p>Inny użytkownik nie posiada wybranej roli (np. Administrator na Rys 2.26)</p> <ol style="list-style-type: none"> 2a. Pole wyboru zostaje zaznaczone 3a. Aplikacja przypisuje wybraną rolę innemu użytkownikowi.

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



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

Źródło: opracowanie własne

Rys. 2.29: Formularz z ustawieniem daty, do której ma być ustawiona 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"> 1. Użytkownik klika przycisk „Artykuły” znajdujący się w panelu nawigacji strony (Rys. 2.14), 2. Aplikacja wyświetla widok związany z wyszukiwaniem artykułów, 3. Użytkownik uzupełnia kryteria wyszukiwania, 4. Użytkownik klika przycisk „Szukaj”, 5. Serwis wyszukuje artykuły, 6. Aplikacja wyświetla listę wyszukanych artykułów (Rys. 2.31), przy czym własne artykuły są oznaczone niebieskim prostokątem.

IT Tech

Technologie IT

Kamil Dywan

Artykuły

Technologie

Wyszukiwanie artykułów

Dodaj artykuł

Tytuł

Java

Autor

kamil.dywan

Kategoria technologii

Język programowania v

Technologia

Java v

Dostawca technologii

Oracle v

Data powstania

01.01.1995

01.07.2000

Data modyfikacji

01.01.2022

01.12.2022

Szukaj

Sortowanie v

<

1

2

...

8

>

Java - poradnik

Java

Język progr.

Adam Nowak

21:00, 01.01.2022

★ 50%

Podstawy Javy

Java

Język progr.

Kamil Dywan

20:30, 04.03.2022

★ 62%

Wersje Javy

Java

Język progr.

Kamil Dywan

22:00, 02.05.2022

★ 80%

Java - strumienie

Java

Język progr.

Michał Nowak

19:30, 06.07.2022

★ 98%

Sortowanie v

<

1

2

...

8

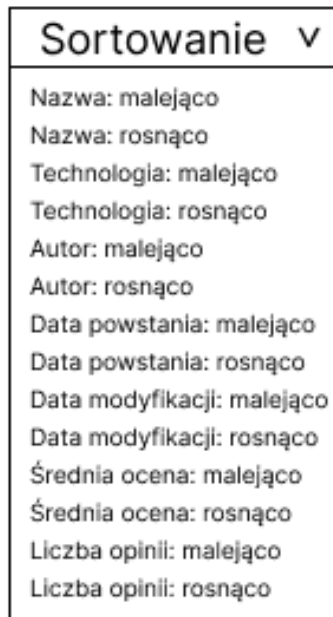
>

Rys. 2.31: Panel do wyszukiwania artykułów z podanymi kryteriami wyszukiwania oraz listą wyszukiwanych 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"> 1. Użytkownik klika na rozsuwaną listę „Sortowanie”, 2. Użytkownik wybiera jedną z opcji sortowania (Rys. 2.32), 3. Aplikacja sortuje artykuły według wybranej opcji sortowania, 4. Aplikacja wyświetla listę posortowanych artykułów, przy czym własne artykuły są oznaczone niebieskim prostokątem.



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"> 1. Użytkownik klika na jeden z wyszukanych artykułów, 2. Aplikacja wyświetla zawartość artykułu oraz komentarze dotyczące tego artykułu (Rys. 2.33).

IT Tech

Technologie IT

Kamil Dywan

Artykuły

Technologie

Kamil Dywan

Status: **Opublikowany**


Edytuj

Usuń

Data utworzenia: 14.07.2022, 00:32:45

Data modyfikacji: 14.07.2022, 00:32:45

Technologia → Język programowania → Java



Dostawca: Oracle

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 zainstalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.

Podstawy Javy

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 kompilacji źródeł kodu w nim napisanych, używana jest maszyna wirtualna nazywana 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.

...

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. Przede wszystkim język jest wykorzystywany w aplikacjach Webowych jako backend oraz aplikacjach mobilnych. W przypadku zastosowań biznesowych język jest wykorzystywany przede wszystkim w rozwiązaniach wysokopoziomowych przez korporacje i duże firmy. Język ten z powodu używanej maszyny wirtualnej jest wolniejszy niż np. C++, dlatego też jest najchętniej 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

★★★★☆ 75%

Komentarze

Opinie

Opis technologii

Dodaj komentarz

Kamil Dywan

15.07.2022, 00:32:45

:

Dobry artykuł

Dodaj komentarz

Rys. 2.33: Panel z zawartością artykułu

Źródło: opracowanie własne

Kamil Dywan: Aplikacja internetowa o technologiach programistycznych

47

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

IT Tech

Technologie IT

Kamil Dywan

Artykuły	Technologie
----------	-------------

Tytuł

Podstawy Javy

Kategoria

Java

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 kompilacji źródeł kodu w nim napisanych, używana jest maszyna wirtualna nazywana 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

Zapisz

Anuluj

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

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

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

Nazwa:	Wyświetlenie listy przypisanych artykułów
Cel:	Wyświetlenie listy artykułów przypisanych 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"> 1. Użytkownik klika przycisk „Weryfikacja artykułów” znajdujący się w menu głównym aplikacji (Rys. 2.35), 2. Aplikacja wyświetla panel z listą artykułów przypisanych użytkownikowi do weryfikacji (Rys. 2.36),

IT Tech Technologie IT		Kamil Dywan 
Artykuły	Technologie	Weryfikacja artykułów

Rys. 2.35: Nagłówek strony recenzenta

Źródło: opracowanie własne

IT Tech Technologie IT

Kamil Dywan

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:	<ol style="list-style-type: none"> 1. Użytkownik klika na jeden z artykułów, 2. Serwis wyświetla formularz do przeprowadzenia weryfikacji artykułu (Rys. 2.37).

IT Tech

Technologie IT

Kamil Dywan

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 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 kompilacji źródeł kodu w nim napisanych, używana jest maszyna wirtualna nazywana JVM (ang. Java Virtual Machine).

- 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.

...
- 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. Przede wszystkim język jest wykorzystywany w aplikacjach Webowych jako backend oraz aplikacjach mobilnych. W przypadku zastosowań biznesowych język jest wykorzystywany przede wszystkim w rozwiązaniach wysokopoziomowych przez korporacje i duże firmy. Język ten z powodu używanej maszyny wirtualnej jest wolniejszy niż np. C++, dlatego też jest najchętniej wykorzystywany w aplikacjach wysokopoziomowych.

Platforma

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

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

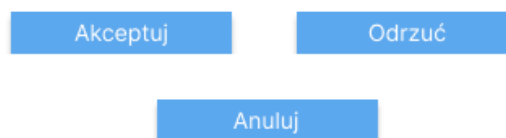
Tab. 2.17: Opis przypadku użycia - odrzucenie artykułu

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"> 1. Użytkownik przegląda cały artykuł, 2. Użytkownik uzupełnia uzasadnienie odrzucenia artykułu (Rys. 2.39), 3. Użytkownik klika przycisk „Odrzuć”, 4. Serwis sprawdza, czy użytkownik uzupełnił pole tekstowe, 5. Po pozytywnej weryfikacji, artykuł zostaje odrzucony.
Alternatywny przebieg:	<p>Użytkownik anuluje akceptację artykułu</p> <ol style="list-style-type: none"> 2a. Użytkownik klika przycisk „Anuluj”, 3a. Serwis wyłącza formularz do odrzucenia artykułu.
Alternatywny przebieg:	<p>Użytkownik nie uzupełnił pola tekstowego</p> <ol style="list-style-type: none"> 5b. Serwis ustawia obramowanie nieuzupełnionego pola tekstowego na czerwono oraz umieszcza pod tym polem komunikat o konieczności jego uzupełnienia, 6b. Powrót do punktu 2.

Wiadomość zwrotna dotycząca treści artykułu

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



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"> 1. Użytkownik klika na jeden z wyszukanых artykułów, 2. Aplikacja wyświetla zawartość wybranego artykułu (Rys. 2.40), 3. Użytkownik klika przycisk „Edytuj”, 4. Serwis wyświetla formularz do edycji artykułu (Rys. 2.34) 5. Użytkownik wypełnia wszystkie pola w formularzu, 6. Użytkownik klika przycisk „Zapisz”, 7. Aplikacja przeprowadza walidację danych pod względem wymaganych pól oraz narzuconych rozmiarów i formatów danych, 8. Po pozytywnej walidacji danych, aplikacja edytuje artykuł według zmian wprowadzonych przez użytkownika, 9. Aplikacja przedstawia użytkownikowi wygląd zmienionego artykułu.
Alternatywny przebieg:	<p>Użytkownik anuluje edytowanie artykułu</p> <ol style="list-style-type: none"> 5a. Użytkownik klika przycisk „Anuluj”, 6a. Użytkownik zostaje przekierowany do strony z wyszukiwaniem artykułów (Rys. 2.31).
Alternatywny przebieg:	<p>Użytkownik nie uzupełnia wszystkich pól formularza podczas edytowania artykułu</p> <ol style="list-style-type: none"> 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.
Alternatywny przebieg:	<p>Usunięcie artykułu</p> <ol style="list-style-type: none"> 3c. Użytkownik klika 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).

Alternatywny przebieg:	Anulowanie usunięcia artykułu 5ca. Użytkownik wybiera opcję „Nie” albo kilka w obszar poza formularzem, 6ca. Artykuł nie zostaje usunięty.
------------------------	--

IT Tech

Technologie IT

Kamil Dywan

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

Kamil Dywan

Status: **Opublikowany**

Edytuj

Usuń

Data utworzenia: 14.07.2022, 00:32:45

Data modyfikacji: 14.07.2022, 00:32:45

Technologia → Język programowania → Java



Dostawca: Oracle

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 zainstalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.

Podstawy Javy

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 kompilacji źródeł kodu w nim napisanych, używana jest maszyna wirtualna nazywana 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.

...

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. Przede wszystkim język jest wykorzystywany w aplikacjach Webowych jako backend oraz aplikacjach mobilnych. W przypadku zastosowań biznesowych język jest wykorzystywany przede wszystkim w rozwiązaniach wysokopoziomowych przez korporacje i duże firmy. Język ten z powodu używanej maszyny wirtualnej jest wolniejszy niż np. C++, dlatego też jest najchętniej 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

★★★★☆

75%

Komentarze

Opinie

Opis technologii

Dodaj komentarz

Kamil Dywan

15.07.2022, 00:32:45

:

Dobry artykuł

Dodaj komentarz

Rys. 2.40: Panel z zawartością artykułu widziany z perspektywy administratora

Źródło: opracowanie własne

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"> 1. Użytkownik przewija artykuł na sam dół, 2. Użytkownik klika przycisk „Komentarze”, 3. Wyświetlenie komentarzy o artykule (Rys. 2.41).



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"> 1. Użytkownik klika przycisk „Dodaj komentarz”, 2. Aplikacja wyświetla formularz do utworzenia komentarza, 3. Użytkownik dodaje tekst komentarza (Rys. 2.42), 4. Użytkownik klika przycisk „Zapisz”, 5. Aplikacja sprawdza, czy użytkownik podał tekst komentarza, 6. Po pozytywnej walidacji komentarza, aplikacja dodaje komentarz do artykułu.

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

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

Zapisz Anuluj

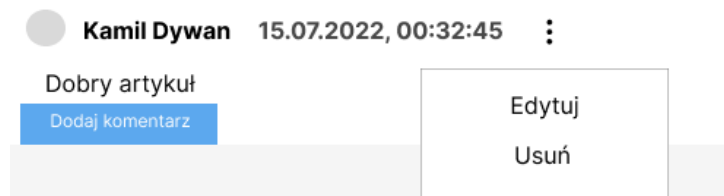
Rys. 2.42: Formularz dodania albo edytowania komentarza o artykule

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:	Autor komentarza o artykule
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"> 1. Użytkownik klika trzy kropki przy jednym z komentarzy, 2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranego komentarza (Rys. 2.43), 3. Użytkownik klika przycisk „Edytuj”, 4. Aplikacja wyświetla formularz do edytowania komentarza (Rys. 2.42), 5. Użytkownik edytuje tekst komentarza, 6. Użytkownik klika przycisk „Zapisz”, 7. Aplikacja sprawdza, czy użytkownik podał tekst komentarza, 8. Po pozytywnej walidacji komentarza, aplikacja edytuje komentarz do artykułu.

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



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

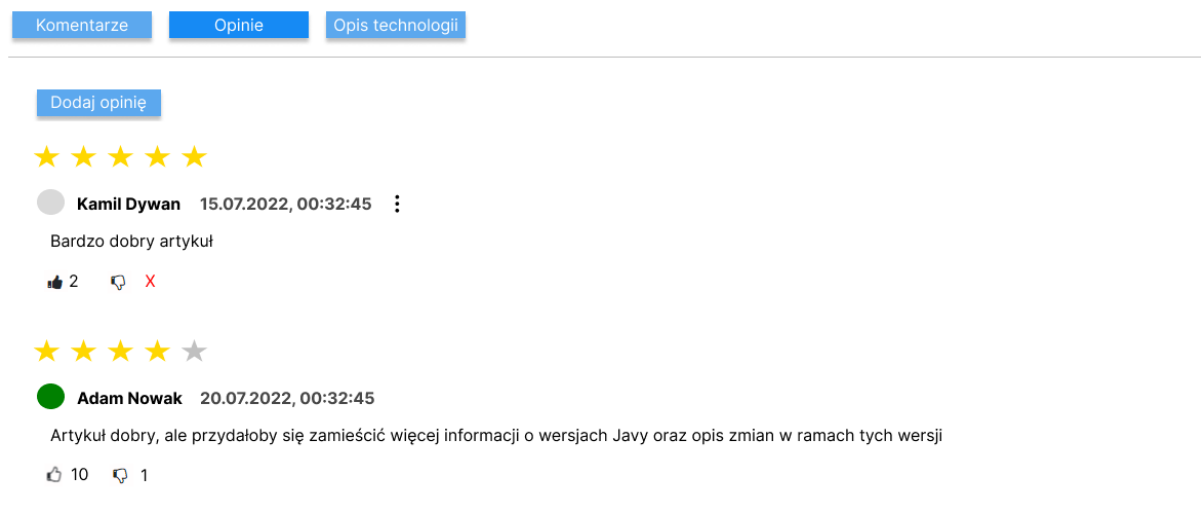


Rys. 2.44: Lista komentarzy o artykule widziana z perspektywy administratora

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"> 1. Użytkownik przewija artykuł na sam dół, 2. Użytkownik klika na przycisk „Opinie”, 3. Aplikacja wyświetla opinie o artykule (Rys. 2.45).



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. 2.45)
Warunki końcowe:	Dodanie opinii o artykule
Przebieg:	<p>Dodanie opinii</p> <ol style="list-style-type: none"> 1. Użytkownik klika przycisk „Dodaj opinię”, 2. Aplikacja wyświetla formularz do utworzenia opinii, 3. Użytkownik dodaje tekst opinii oraz wystawia ocenę (Rys. 2.46), 4. Użytkownik klika przycisk „Zapisz”, 5. Aplikacja sprawdza, czy użytkownik podał tekst opinii oraz ocenę, 6. Po pozytywnej walidacji, aplikacja dodaje opinię.
Alternatywny przebieg:	<p>Użytkownik anuluje tworzenie opinii</p> <ol style="list-style-type: none"> 3a. Użytkownik klika przycisk „Anuluj” na formularzu do tworzenia opinii, 4a. Wyłączenie formularza do dodania opinii.

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”,</p> <p>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: Formularz dodania albo edytowania opinii o artykule

Źródło: opracowanie własne

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

Nazwa:	Zarządzanie własną opinią o artykule
Cel:	Modyfikacja lub usunięcie własnej opinii do artykułu
Aktorzy:	Autor opinii o artykule
Warunki początkowe:	Wyświetlone opinie o artykule (Rys. 2.45)
Warunki końcowe:	Modyfikacja lub usunięcie własnej opinii do artykułu
Przebieg:	<p>Edytowanie opinii</p> <ol style="list-style-type: none"> 1. Użytkownik klika trzy kropki przy jednej z opinii, 2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranej opinii 3. Użytkownik klika przycisk „Edytuj”, 4. Aplikacja wyświetla formularz do edytowania opinii (Rys. 2.46), 5. Użytkownik edytuje opinię, 6. Użytkownik klika przycisk „Zapisz”, 7. Aplikacja sprawdza, czy użytkownik podał tekst opinii oraz ocenę, 8. Po pozytywnej walidacji, aplikacja edytuje opinię.
Alternatywny przebieg:	<p>Użytkownik anuluje edytowanie opinii</p> <ol style="list-style-type: none"> 5a. Użytkownik klika przycisk „Anuluj” na formularzu do edytowania opinii, 6a. Wyłączenie formularza do edytowania opinii.

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

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"> 1. Użytkownik klika trzy kropki przy jednej z opinii, 2. Aplikacja wyświetla listę działań, które mogą zostać wykonane w ramach wybranej opinii 3. Użytkownik klika przycisk „Usuń”, 4. Aplikacja wyświetla formularz z zapytaniem „Czy na pewno ta opinia powinna zostać usunięta?”, 5. Użytkownik wybiera opcję „Tak”, 6. Serwis usuwa opinię.
Alternatywny przebieg:	<p>Użytkownik anuluje usuwanie opinii</p> <p>5a. Użytkownik wybiera opcję „Nie”,</p> <p>6a. Opinia nie zostaje usunięta.</p>



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. 2.45)
Warunki końcowe:	Wyświetlenie opinii z zaktualizowanym stanem akceptacji pozytywnych i negatywnych
Przebieg:	<p>Dodanie pozytywnej akceptacji</p> <ol style="list-style-type: none"> 1. Użytkownik klika przy jednej z opinii kciuk w górę, 2. Aplikacja zwiększa o 1 liczbę pozytywnych akceptacji wybranej opinii, 3. Aplikacja zmienia kolor kciuka w górę na czarny, 4. Aplikacja wyświetla opinię z zaktualizowanym stanem akceptacji pozytywnych i negatywnych.
Alternatywny przebieg:	<p>Dodanie negatywnej akceptacji</p> <ol style="list-style-type: none"> 1a. Użytkownik klika przy jednej z opinii kciuk w dół, 2a. Aplikacja zwiększa o 1 liczbę negatywnych akceptacji wybranej opinii, 3a. Aplikacja zmienia kolor kciuka w dół na czarny, 4a. Aplikacja wyświetla opinię z zaktualizowanym stanem akceptacji pozytywnych i negatywnych.

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. 2.45)
Warunki końcowe:	Wyświetlenie opinii z zaktualizowanym stanem akceptacji pozytywnych i negatywnych
Przebieg:	<ol style="list-style-type: none"> 1. Użytkownik klika w czerwony znak „X” przy akceptacjach wybranej opinii, 2. Aplikacja usuwa akceptację opinii, 3. Aplikacja zmienia kolor ustawionego kciuka na biały, 4. Aplikacja wyświetla opinię z zaktualizowanym stanem akceptacji pozytywnych i negatywnych.

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"> 1. Użytkownik klika przycisk „Technologie” znajdujący się w panelu nawigacji strony (Rys. 2.14), 2. Aplikacja wyświetla panel do wyszukiwania technologii, 3. Użytkownik uzupełnia kryteria wyszukiwania, 4. Użytkownik klika przycisk „Szukaj”, 5. Serwis wyszukuje technologii, 6. Aplikacja wyświetla listę wyszukanych technologii (Rys. 2.48).

IT Tech

Technologie IT

Kamil Dywan

Artykuły

Technologie

Wyszukiwanie technologii

Nazwa

Java

Kategoria

Język programowania

Zastosowania

Backend

Dostawca

Oracle

Data powstania

02.03.1950

03.05.2022

Data modyfikacji

09.07.2020

28.08.2022

Szukaj

Sortowanie

<

1

2

...

8

>

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

<

1

2

...

8

>

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

Źródło: opracowanie własne

Tab. 2.30: Opis przypadku użycia - sortowanie technologii

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

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:	<ol style="list-style-type: none"> 1. Użytkownik wyszukuje technologie (Tab. 2.29) 2. Użytkownik klika na jedną z wyszukanych technologii, 3. Aplikacja wyświetla opis technologii (Rys. 2.50).
Alternatywny przebieg:	<p>Użytkownik wyświetla opis technologii poprzez artykuł</p> <ol style="list-style-type: none"> 1a. Użytkownik wyszukuje artykuły (Tab. 2.9), 2a. Użytkownik wyświetla zawartość artykułu (Tab. 2.11) zawierającą na początku opis technologii, której artykuł dotyczy.

IT Tech

Technologie IT

Kamil Dywan

Artykuły

Technologie

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 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 zainstalowane 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 albo usunięcie istniejącej technologii
Aktorzy:	Recenzent
Warunki początkowe:	Wyświetlony panel do wyszukiwania technologii (Rys. 2.51)
Warunki końcowe:	Dodanie nowej technologii, modyfikacja albo usunięcie istniejącej technologii

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

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

IT Tech

Technologie IT

Kamil Dywan

Artykuły

Technologie

Wyszukiwanie technologii

Dodaj technologię

Nazwa

Java

Kategoria

Język programowania

Zastosowania

Backend

Dostawca

Oracle

Data powstania

02.03.1950

03.05.2022

Data modyfikacji

09.07.2020

28.08.2022

Szukaj

Sortowanie

<

1

2

...

8

>

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

<

1

2

...

8

>

Rys. 2.51: Panel do wyszukiwania technologii z podanymi kryteriami wyszukiwania oraz listą wyszukiwanych technologii widziany z perspektywy recenzenta

Źródło: opracowanie własne

IT Tech

Technologie IT

Kamil Dywan

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

Nazwa

Java *

Kategoria

Języki progr. v *

Dostawca

Oracle

Data ostatniego wydania

01.01.1996

Data ostatniego wydania

22.03.2022

Ikona

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 zainstalowane JRE (kod bajtowy klas standardowych razem z maszyną wirtualną). Obecnie Java jest wykorzystywana przede wszystkim jako backend w aplikacjach webowych i mobilnych.

Zapisz

Anuluj

Rys. 2.52: Panel z formularzem dodania albo modyfikacji technologii

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:	Minęły 24 godziny od ostatniego wyszukania artykułów z przeterminowaną weryfikacją
Warunki końcowe:	Zwrócenie artykułów z przeterminowaną weryfikacją

Przebieg:	<ol style="list-style-type: none"> 1. Wyszukanie artykułów z przedawnioną weryfikacją, 2. Zapisanie artykułów z przedawnioną weryfikacją do kolejki artykułów do weryfikacji (jeśli nie są jeszcze w tej kolejce), 3. Zwrócenie artykułów znajdujących się w kolejce artykułów do weryfikacji.
-----------	---

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

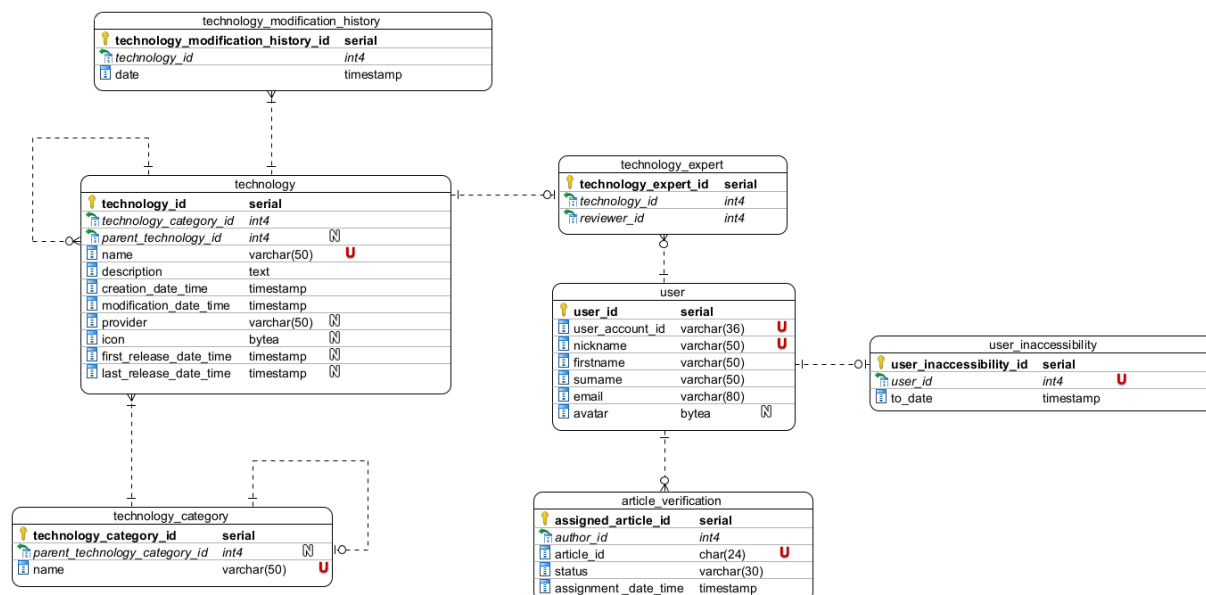
Nazwa:	Przypisanie recenzentowi artykułu do weryfikacji
Cel:	Przypisanie recenzentowi artykułu do weryfikacji
Aktorzy:	System
Warunki początkowe:	Zalogowany użytkownik utworzył artykuł albo autor artykułu dokonał modyfikacji istniejącego artykułu
Warunki końcowe:	Przypisanie wybranemu recenzentowi artykułu do weryfikacji albo oczekiwanie na dostępnego recenzenta
Przebieg:	<p>Jest przynajmniej jeden dostępny recenzent</p> <ol style="list-style-type: none"> 1. Przypisanie wybranemu dostępnemu recenzentowi artykułu do weryfikacji,
Alternatywny przebieg:	<p>Nie ma dostępnych recenzentów</p> <ol style="list-style-type: none"> 1a. System zapisuje artykuł do kolejki artykułów do weryfikacji, 2a. System sprawdza co 24 godziny, czy pojawił się dostępny recenzenta, 3a. Nie ma dalej dostępnych recenzentów.
Alternatywny przebieg:	<p>Pojawił się dostępny recenzent</p> <ol style="list-style-type: none"> 3aa. Powrót do punktu 1.

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:	Minał termin końcowy weryfikacji artykułu
Warunki końcowe:	Przypisanie innemu recenzentowi artykułu do weryfikacji albo oczekiwanie na pojawienie się dostępnego recenzenta
Przebieg:	<ol style="list-style-type: none"> 1. Przypisanie artykułu do weryfikacji innemu recenzentowi.

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 2.53.



Rys. 2.53: 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 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      "$oid":"634308ab8f35533467caa4f7"
4    },
5    "authorId":27,
6    "technologyId":1,
7    "title":"Podstawy Javy1",
8    "content":"<p style=\"text-align: justify;\">Java jest to wsp&oacute;
      ↳  łbieżny, opaty na klasach i mocno typowany obiektowy język
      ↳  programowania...",
9    "status":"NEW",
10   "creationDate":{
11     "$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 }
```

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

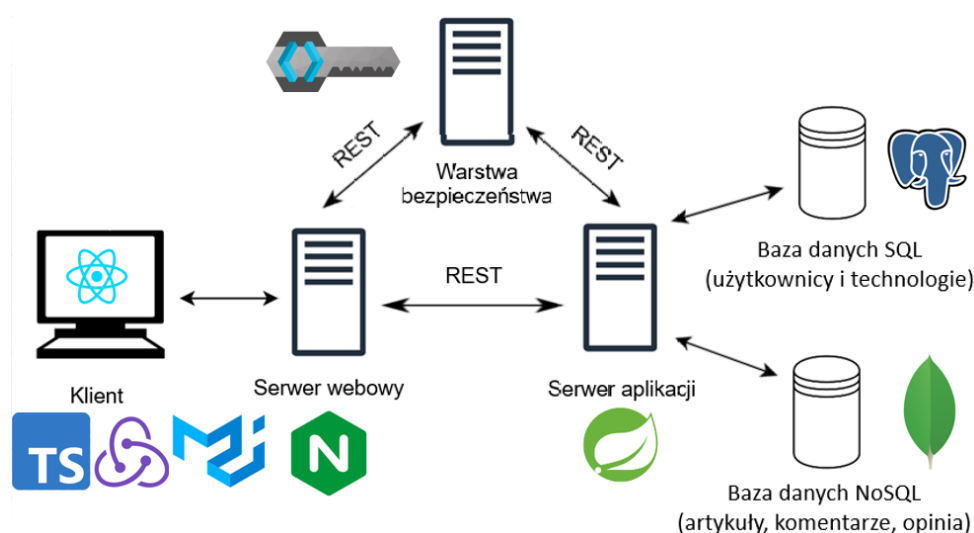
3.1. Technologie i narzędzia

W celu zaimplementowania zaprojektowanego systemu dokonano wyboru technologii i narzędzi. Przy ich wyborze kierowano się głównie ich odpowiednością do realizowanego systemu, popularnością oraz własnym doświadczeniem uzyskanym w trakcie studiów. Poszukiwano przede wszystkim darmowych rozwiązań, gdyż realizowany w ramach tej pracy projekt jest niekomercyjny. 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 Managment 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. Javę 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 frameworka 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 unikać niepotrzebnego powielania kodu.

JavaScript jest językiem słabo typowanym i przez to w trakcie kompilacji 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 autoryzację. 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.

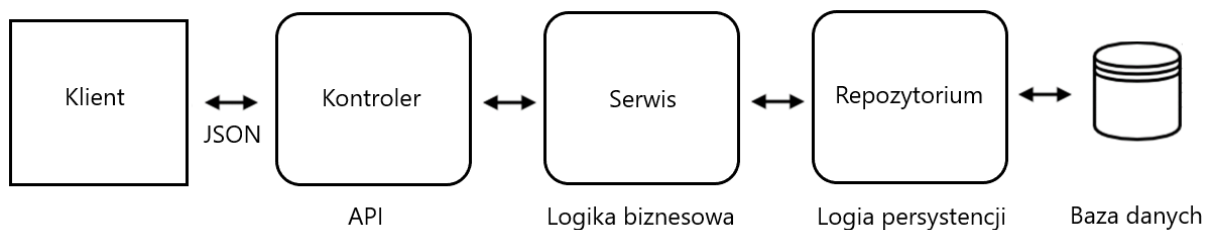
3.1.2. Narzędzia

Użyto następujących narzędzi:

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

3.2. Aplikacja serwerowa

Aplikacja serwerowa jest dosyć złożonym systemem, gdyż odpowiada za wiele różnych funkcji np. udostępnianie endpointów klientowi, czy zarządzanie bazą danych. Z powodu tej złożoności niezbędnym było podzielenie aplikacji serwerowej na warstwy (Rys. 3.2). Każda warstwa odpowiada za odrębne funkcje. Komunikacja może się odbywać tylko pomiędzy sąsiednimi warstwami.



Rys. 3.2: Architektura aplikacji serwerowej

Źródło: opracowanie własne

Aplikacja serwerowa wykorzystuje mapowanie obiektowo-relacyjne (ang. *ORM*) poprzez framework *Hibernate* i dzięki temu możliwe jest wygodne operowanie na obiektach zamiast ręcznego wywoływania mniej czytelnych i mniej intuicyjnych zapytań do bazy danych.

Warstwa bazy danych odpowiada za zarządzanie bazą danych. Warstwa ta operuje na tzw. encjach (modelach), czyli odpowiednio oznaczonych klasach Javy, które odpowiadają tabelom albo dokumentom w bazie danych. W zaprojektowanym systemie można wyróżnić dwa rodzaje encji: przechowywane przez bazę PostgreSQL oraz MongoDB.

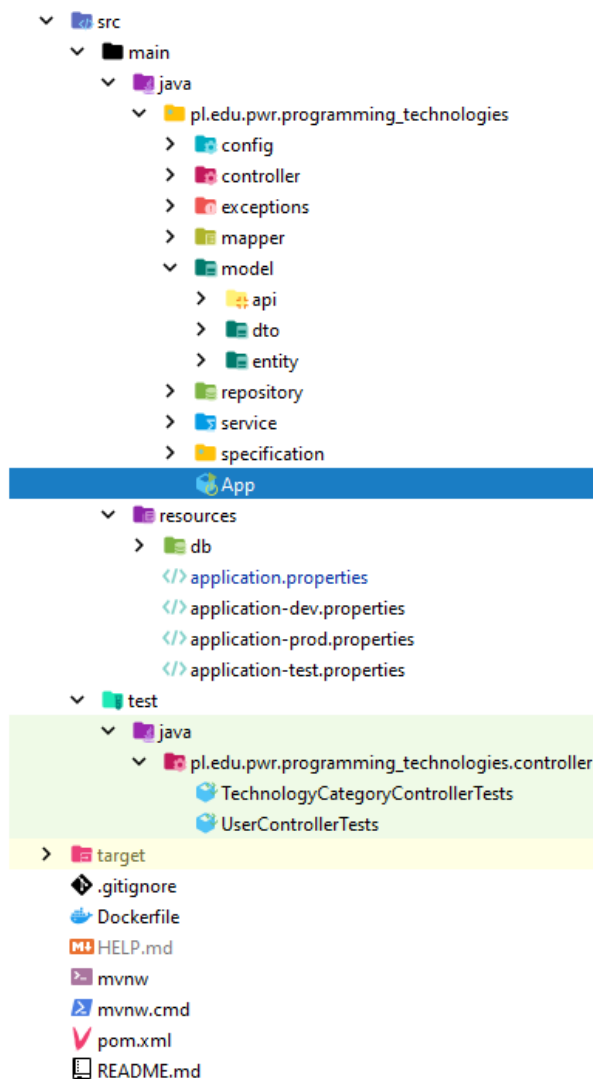
Repozytorium jest ściśle powiązane z warstwą bazy danych. Warstwa ta nakłada na bazę danych abstrakcję (logikę persystencji), która pozwala programiście na wywoływanie metod zamiast ręcznego pisania zapytań do bazy danych. Do repozytorium jest przypisana jedna encja wraz z jej kluczem podstawowym i repozytorium operuje jedynie na tej jednej encji. Tak samo jak w przypadku modeli, można wyróżnić dwa rodzaje repozytoriów: operujące na bazie PostgreSQL oraz MongoDB.

Warstwa serwisu implementuje logikę biznesową systemu. Serwisy są to odpowiednio oznaczone klasy Javy, których metody implementują poszczególne funkcjonalności systemu od strony backendu. Serwisy korzystają z repozytoriów.

Warstwa kontrolera odpowiada za udostępnianie klientowi usług (np. utworzenie artykułu) dostarczanych przez serwisy. Kontroler udostępnia grupę endpointów, z których każdy z nich dostarcza jedną z usług. Klient może skorzystać z danej usługi poprzez wysłanie żądania na endpoint powiązany z tą usługą.

Warstwa kliencka jest to aplikacja kliencka, która wysyła żądania do kontrolera, w celu skorzystania z udostępnionych przez niego usług.

Struktura aplikacji serwerowej jest przedstawiona na rysunku 3.3



Rys. 3.3: Struktura projektu aplikacji serwerowej

Źródło: opracowanie własne

Struktura aplikacji serwerowej jest zgodna z przedstawioną architekturą (Rys. 3.2). Backend jest oparty na narzędziu automatyzującym budowanie aplikacji Maven.

Konfiguracja aplikacji (np. link do bazy danych, czy numer portu na jakim działa aplikacja) jest zapisana w plikach o rozszerzeniu `properties`. Można wyróżnić 3 różne profile konfiguracyjne aplikacji: lokalny, produkcyjny oraz do testowania. Profil lokalny jest wykorzystywany w trakcie rozwoju aplikacji i w konfiguracji tej używane są lokalne bazy danych oraz jest możliwość modyfikowania struktury tabel (schematów dokumentów). Profil produkcyjny jest wykorzystywany przez wdrożoną aplikację i w konfiguracji tej używane są zdalne bazy danych oraz nie ma możliwości modyfikowania struktury tabel (schematów dokumentów). Profil do testowania jest wykorzystywany do testów aplikacji i w konfiguracji tej używana jest baza danych h2, która jest przechowywana w pamięci.

Przykładowy profil lokalny jest przedstawiony na listingu 3.1.

Listing 3.1: Profil lokalny

```

server.port=9000
spring.datasource.url = jdbc:postgresql://localhost:5432/
    ↪ programming_technologies
spring.datasource.username = programming_technologies
spring.datasource.password = postgres
  
```

```

spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=technologie-it
spring.data.mongodb.username=programming_technologies
spring.data.mongodb.password=mongodb

spring.jpa.hibernate.ddl-auto = update

keycloak.auth-server-url=http://localhost:8080
frontend.url=http://localhost:3000

```

Fragment przykładowej encji technologii przechowywanej przez bazę PostgreSQL przedstawiono na listingu 3.2.

Listing 3.2: Encja technologii

```

@Entity
@Table(name="TECHNOLOGIES")
public class TechnologyEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "technology_id")
    private Integer id;

    @Column(name = "name", nullable = false, unique = true)
    private String name;

    @Column(name = "description", nullable = false)
    private String description;

    @Column(name = "creation_date_time", nullable = false)
    private LocalDateTime creationDateTime;

    ...

    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    @JoinColumn(name = "technology_category_id")
    private TechnologyCategoryEntity technologyCategoryEntity;
}

```

Aby klasy były interpretowane jako encje i mogły być przechowywane w bazie PostgreSQL, należy nadać takiej klasie adnotację `Entity`. Wymagane jest również, aby encje posiadały zdefiniowane klucze podstawowe (`Id`). Klucze podstawowe są generowane automatycznie przez PostgreSQL i oznaczono to za pomocą wartości `Identity`. Założono, że encje będą oparte są na zdefiniowanych wcześniej schematach bazy danych, dlatego każdy z atrybutów encji należało dostosować do odpowiadających im pól na schemacie za pomocą odpowiedniej konfiguracji adnotacji `Column`. Przykładowo atrybut `creationDateTime` odpowiada polu o nazwie `creation_date_time` po stronie bazy danych. Ważnym elementem w przypadku stosowania ORM jest definiowanie relacji. Użyte technologie pozwalają na proste definiowanie relacji między encjami, które ogranicza się do zastosowania kilku adnotacji np. relacja wiele do jeden między technologiami i kategoriami technologii. W przypadku relacji, można również m.in. zdefiniować kaskadowe aktualizowanie i usuwanie podrzędnej encji (np. relacja jeden do wielu technologii względem kategorii technologii 3.3), co okazuje się być bardzo przydatne.

Listing 3.3: Mapowanie jeden do wielu między technologiami i kategoriami technologii

```

@Transient
@OneToMany(mappedBy="technologyCategoryEntity", fetch = FetchType.LAZY,
    ↪ cascade = CascadeType.ALL, orphanRemoval=true)
private List<TechnologyEntity> technologyEntityList;
}

```

Fragment przykładowej encji artykułu przechowywanej przez bazę MongoDB przedstawiono na listingu 3.4.

Listing 3.4: Encja artykułu

```

@Builder
@Data
@NoArgsConstructor
@AllArgsConstructor
@Document("articles")
public class ArticleEntity {

    ...

    @Id
    @MongoId(FieldType.OBJECT_ID)
    @Field(name = "_id")
    private ObjectId id;

    @NotNull
    @Field(name = "authorId")
    private Integer authorId;

    @NotNull
    @Field(name = "technologyId")
    private Integer technologyId;

    @NotNull
    @Indexed(unique = true)
    @Field(name = "title")
    private String title;

    @NotNull
    @Field(name = "content")
    private String content;

    @NotNull
    @Enumerated(EnumType.STRING)
    @Field(name = "status")
    private Status status;

    ...

    @Min(0)
    @Field(name = "averageRating")
    private Double averageRating;
}

```

Aby klasy były interpretowane jako encje i mogły być przechowywane w bazie MongoDB, należy nadać takiej klasie adnotację `Document`. Tak samo jak w przypadku PostgreSQL, wymagane jest aby encje posiadały zdefiniowane klucze podstawowe, ale klucze te powinny być typu `ObjectId` oraz muszą być one jeszcze oznaczone adnotacją `MongoId`. Klucze podstawowe są również generowane automatycznie przez MongoDB. Podobnie i w tym przypadku encje są

oparte na zdefiniowanych wcześniej schematach bazy danych i każdy z atrybutów encji dostosowano do odpowiadających im pól na schematach, ale służyła do tego adnotacja `Field`, a nie `Column`. Niestety, ale przy użyciu frameworka Spring dla baz NoSQL, nie jest możliwe tworzenie relacji opartych na ORM między encjami. Rozwiązaniem tego problemu jest przechowywanie identyfikatorów powiązanych encji (np. `authorId`) oraz zapewnienie integralności danych po stronie backendu.

Fragment przykładowego repozytorium technologii operującego na bazie PostgreSQL przedstawiono na listingu ??.

Listing 3.5: Repozytorium technologii

```
@Repository
public interface TechnologyRepository extends JpaRepository<
    ↪ TechnologyEntity, Integer> {

    List<TechnologyEntity> findAllByTechnologyCategoryId(Integer
        ↪ technologyCategoryId);
    List<TechnologyEntity> findAllByProviderContainsIgnoreCase(String
        ↪ provider);

    default List<TechnologyEntity> findAllHavingTechnologyCategoryIdInTree(
        ↪ Integer technologyCategoryId){
        return findAll()
            .stream()
            .filter(t -> {
                Optional<TechnologyCategoryEntity> opt = Optional.
                    ↪ ofNullable(t.getTechnologyCategoryEntity());
                while(opt.isPresent()) {
                    if(opt.get().getId() == technologyCategoryId){
                        return true;
                    }
                    opt = Optional.ofNullable(opt.get().
                        ↪ getParentTechnologyCategoryEntity());
                }
                return false;
            })
            .collect(Collectors.toList());
    }
}
```

Aby interfejsy były interpretowane jako repozytoria, muszą być one oznaczone adnotacją `Repository`. Dziedziczenie repozytoriów po interfejsie `JpaRepository`, pozwala na uzyskanie dostępu do kilku wygenerowanych metod np. metodę do pobierania wszystkich encji. Inną funkcją tego dziedziczenia jest możliwość wygenerowania metod na podstawie samych deklaracji. Przykładowo deklaracja `findAllByTechnologyCategoryId` pozwoli na wygenerowanie metody zwracającej listę technologii o danej kategorii.

Fragment przykładowego repozytorium artykułów operującego na bazie MongoDB przedstawiono na listingu ??.

Listing 3.6: Repozytorium technologii

```
@Repository
public interface ArticleRepository extends MongoRepository <ArticleEntity,
    ↪ ObjectId> {

    boolean existsByTitleIgnoreCase(String title);
    List<ArticleEntity> findAllByStatusOrStatus(ArticleEntity.Status
        ↪ status1, ArticleEntity.Status status2);
}
```

Podobnie jest w przypadku repozytoriów operujących na bazie MongoDB. Jediną różnicą jest to, że repozytorium powinno dziedziczyć po interfejsie `MongoRepository`.

Fragment przykładowego serwisu artykułów przedstawiono na listingu ??.

Listing 3.7: Repozytorium technologii

```
@Slf4j
@Service
@RequiredArgsConstructor
public class ArticleServiceImpl implements ArticleService {

    private final ArticleRepository articleRepository;
    private final CommentRepository commentRepository;
    private final UserRepository userRepository;
    private final TechnologyRepository technologyRepository;

    ...

    @Override
    public ArticleEntity getArticleById(ObjectId articleId) throws
        ↪ EntityNotFoundException{

        Optional<ArticleEntity> foundArticleOpt = articleRepository.
            ↪ findById(articleId);

        if(foundArticleOpt.isEmpty()){
            throw new EntityNotFoundException("Nie istnieje artykuł o takim
                ↪ id");
        }

        return foundArticleOpt.get();
    }

    ...
}
```

Aby interfejsy były interpretowane jako repozytoria, muszą być one oznaczone adnotacją `Service`. Serwis artykułów (listing ??) korzysta z repozytoriów artykułów, komentarzy, użytkowników oraz technologii. W serwisach zaimplementowana jest obsługa wyjątków i np. w przypadku nie podania id artykułu w metodzie `getArticleById`, zostanie rzucony wyjątek z komunikatem „Nie istnieje artykuł o takim id”. Wyjątki rzucane przez serwisy są później obsługiwane przez kontrolery.

Fragment przykładowego kontrolera artykułów przedstawiono na listingu ??.

Listing 3.8: Repozytorium technologii

```
@RestController
@CrossOrigin(origins = {"http://localhost:3000", "https://technologie-
    ↪ programistyczne.azurewebsites.net"})
@RequiredArgsConstructor
@RequestMapping(value = "/article")
public class ArticleController {

    private final ArticleService articleService;
    private final CommentService commentService;
    private final CommentMapper commentMapper = CommentMapper.INSTANCE;
    private final ArticleMapper articleMapper = ArticleMapper.INSTANCE;
    private final SearchCriteriaMapper searchCriteriaMapper =
        ↪ SearchCriteriaMapper.INSTANCE;
    private final UserRepository userRepository;
```

```
private final TechnologyRepository technologyRepository;

...

@GetMapping
public ResponseEntity<Page<ArticleDTO>> getAll(Pageable pageable){

    if(pageable == null){
        pageable = Pageable.unpaged();
    }

    Page<ArticleEntity> articleEntityPage = articleService.getAll(
        ↪ pageable);
    List<ArticleDTO> articleDTOList = articleMapper.
        ↪ articleEntityListToArticleDTOList(
            articleEntityPage.getContent(), userRepository,
            ↪ technologyRepository
        );
    Page<ArticleDTO> articleDTOPage = new PageImpl<>(articleDTOList,
        ↪ pageable, articleEntityPage.getTotalElements());

    return ResponseEntity.ok(articleDTOPage);
}

...
}
```

Rozdział 4

Testowanie

Rozdział 5

Wdrożenie

Rozdział 6

Podsumowanie

Literatura

Dodatek A

Instrukcja wdrożeniowa

Dodatek B

Opis załączonej płyty CD/DVD