

PROJEKT ZESPOŁOWY 2

DOKUMENTACJA

Temat projektu: Portal dla przyszłych inżynierów

Zespół 07

Hubert Soroka

Maciej Tymoftejewicz

Daniel Kobiałka

Mateusz Krakowski

12 stycznia 2023

Spis treści

| | | |
|----------|---|-----------|
| 1 | Specyfikacja wymagań | 2 |
| 1.1 | Zadanie biznesowe | 2 |
| 1.2 | Wymagania użytkowe, biznesowe i systemowe | 2 |
| 2 | Definicja architektury | 3 |
| 2.1 | Plan struktury systemu | 3 |
| 2.2 | Link do repozytorium projektu | 3 |
| 3 | Specyfikacja analityczna | 3 |
| 3.1 | Model dziedziny | 3 |
| 3.2 | Słownik pojęć | 4 |
| 3.3 | Model ERD struktury bazodanowej | 4 |
| 4 | Specyfikacja projektowa | 17 |
| 4.1 | Metoda Realizacji | 17 |
| 4.2 | Logiczny model danych | 18 |
| 4.3 | Model obiektowy systemu | 18 |
| 4.4 | Intefrejs użytkownika | 19 |
| 4.5 | Standardy obsługi błędów i sytuacji wyjątkowych | 27 |
| 5 | Specyfikacja testów | 27 |
| 5.1 | Scenariusze testów funkcjonalnych | 27 |
| 6 | Podręcznik użytkownika | 28 |
| 6.1 | Operacje możliwe do wykonania w aplikacji | 28 |
| 7 | Podręcznik administratora | 29 |
| 7.1 | Instrukcja instalacji i konfiguracji systemu | 29 |
| 7.2 | Instrukcja aktualizacji oprogramowania | 29 |
| 7.3 | Instrukcja tworzenia kopii zapasowej | 30 |
| 7.4 | Instrukcja odtwarzania systemu | 30 |
| 8 | Przydatne linki | 30 |
| 8.1 | Link do repozytorium projektu | 30 |
| 8.2 | Link do załączników | 30 |

1 Specyfikacja wymagań

1.1 Zadanie biznesowe

Aplikacja zbierająca i sprawdzająca wiedzę z zakresu studiów licencjackich w postaci scenariuszy testowych (ciągów pytań) na różnych poziomach trudności. System powinien w szczególności odsyłać do niezbędnych źródeł wiedzy: materiałów z wykładów i ćwiczeń, literatury pomocniczej, GITów i innych zasobów sieci.

1.2 Wymagania użytkowe, biznesowe i systemowe

Tematem naszego projektu jest platforma edukacyjna, pozwalająca na dodawanie pytań połączonych z odpowiedzią i źródłem wiedzy (taki zestaw będzie dalej zwany fiszką), przeglądanie fiszek oraz grupowanie ich z użyciem grup i tagów. Użytkownicy muszą się zalogować, aby przeglądać i dodawać nowe fiszki i grupy, które mają ustawienia publiczności.

Funkcjonalności aplikacji:

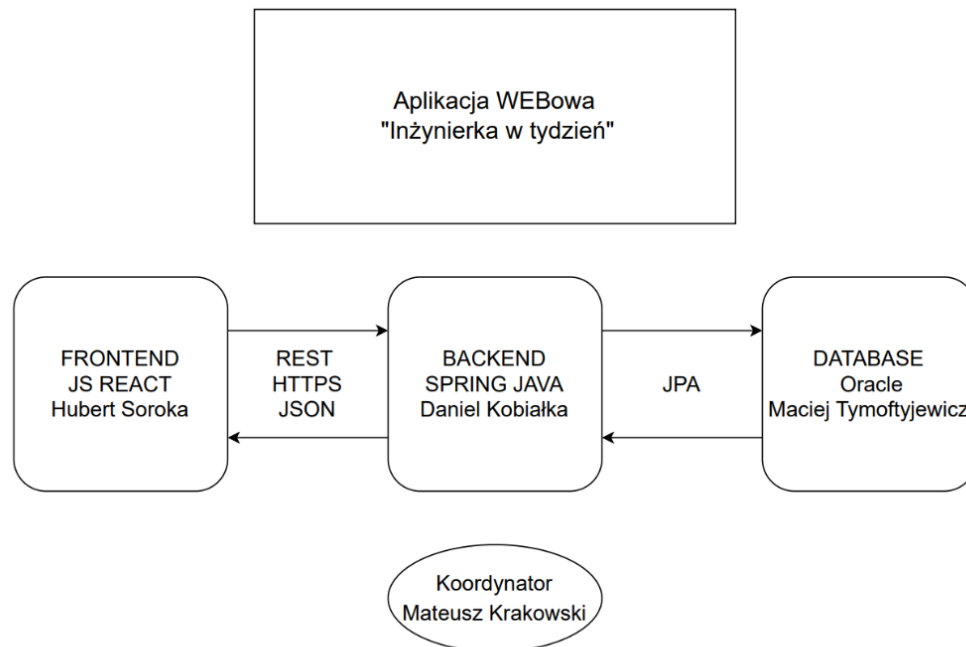
- Założenie konta użytkownika
- Zalogowanie się na konto
- Wylogowanie się z konta
- Tworzenie, edytowanie oraz usuwanie fiszek
- Tworzenie, edytowanie oraz usuwanie grup
- Wyszukanie fiszek i grup po nazwie
- Dodawanie tagów
- Wyszukanie fiszek i grup po tagu
- Dodanie fiszki do grupy (podczas dodawania lub edycji grupy, oraz podczas dodawania lub edycji fiszki)
- Przechodzenie przez tryb nauki dla danej grupy (przeglądanie fiszek)
- Przechodzenie przez tryb testu dla danej grupy (użytkownik widzi pytanie, może udzielić odpowiedzi i po przejściu przez test porównać swoje odpowiedzi do odpowiedzi prawidłowych)

Nasz projekt nie jest nastawiony na czerpanie zysków. Naszym celem jest stworzenie w pełni darmowego portalu dla studentów, nie jest planowane wprowadzanie żadnych sposobów monetyzacji. Należy wspomnieć, że dostępność portalu jest kluczowa w jego działaniu. Aby portal sprawnie działał, muszą z niego korzystać ludzie skłonni poświęcić swój czas na przygotowanie materiałów dydaktycznych.

Zważając na to, że z projektu nie będziemy czerpać korzyści majątkowych, liczymy że uda nam się namówić jakąś organizację nienastawioną na zysk, na udostępnienie nam zasobów serwerowych, aby nasz projekt mógł działać.

2 Definicja architektury

2.1 Plan struktury systemu



Nasza aplikacja składa się z trzech niezależnych od siebie części:

- Frontend - Typescript/React
- Backend - Java/Spring
- Baza danych - Oracle

Baza danych komunikuje się z backendem poprzez interfejs Spring Data JPA, który wykorzystuje driver JDBC. Backend wystawia Restowe API, przez które frontend nawiązuje połączenie. Dane przesyłane Jsonem, protokół http.

2.2 Link do repozytorium projektu

LINK DO REPOZYTORIUM PROJEKTU

3 Specyfikacja analityczna

3.1 Model dziedziny

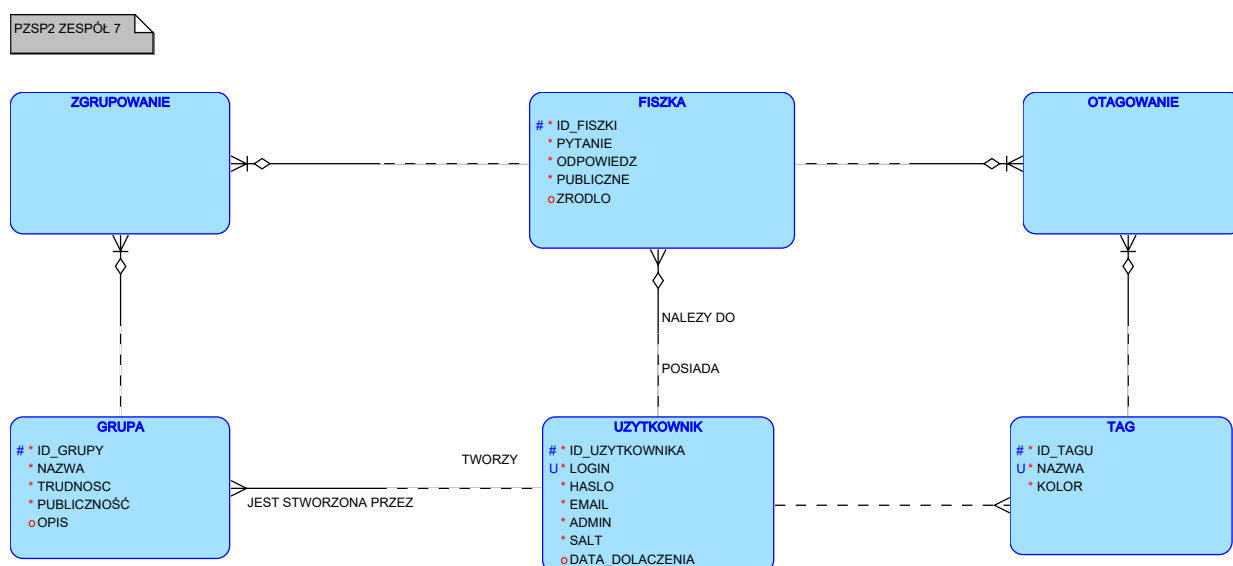
Naszą aplikację tworzymy dla studentów chcących kumulować oraz wykorzystywać zasoby wiedzy. Jako że jesteśmy studentami, możemy szczerze powiedzieć, że najwygodniejszym sposobem na przyswajanie wiedzy, jest właśnie korzystanie z fiszek. Ten dość prosty koncept przenieśliśmy

do świata wirtualnego, aby zamiast stosu małych karteczek mieć wygodne do przeglądania oraz grupowania fiszki w aplikacji.

3.2 Słownik pojęć

- Fiszka – podstawowy element aplikacji, służy do nauki. Składa się z pytania, odpowiedzi oraz źródła.
- Grupa – grupa fiszek tworzona przez użytkowników.
- Użytkownik – osoba korzystająca z aplikacji.
- Tag – słowo kluczowe przypisane do fiszki lub grupy.
- Publiczne, prywatne – zarówno fiszki, jak i grupy dzielone są ze względu na ich widoczność. Zasoby publiczne widoczne są dla wszystkich, prywatne - tylko dla użytkownika, który je dodał.
- Tryb przeglądania – analogia do nauki przy użyciu fizycznych fiszek. Użytkownik przegląda swoje fiszki z możliwością podejrzenia odpowiedzi oraz źródła.
- Scenariusz testowy – alternatywny tryb nauki przy użyciu fiszek. Użytkownik przegląda swoje fiszki przy czym próbuje odpowiadać na pytania. W czasie nauki użytkownik zaznacza, czy odpowiedział dobrze, czy źle. Po zakończeniu scenariusza testowego użytkownik otrzymuje informację zwrotną zawierającą jego wynik w postaci procentowej.
- Trudność – fiszki oraz grupy posiadają trudność, która oznacza poziom zaawansowania materiału, który zawierają.

3.3 Model ERD struktury bazodanowej



Model również znajduje się w załączniku ERD.pdf w lepszej rozdzielczości (Link na końcu tego pliku).

USE CASES

Aktorzy:

- Użytkownik + podtypy

Scenariusze:

- Założenie konta (Niezarejestrowany użytkownik)
- Zalogowanie (Niezalogowany użytkownik)
- Tworzenie fiszki
- Tworzenie grupy
- Udostępnianie grupy fiszek (Użytkownik udostępniający, Użytkownik, dla którego grupa jest udostępniana)
- Wyszukanie grupy po nazwie/tagu
- Wyszukiwanie fiszki
- Edycja grupy
- Edycja fiszki
- Usunięcie fiszki
- Usunięcie grupy
- Dodanie fiszki do grupy

| | | |
|-----------------|--|--|
| Scenariusz 1 | Założenie konta | |
| Opis | Użytkownik chce założyć konto | |
| Aktorzy | Niezarejestrowany użytkownik | |
| Warunki wstępne | - | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę aplikacji |
| | 2 | Użytkownik klika przycisk „Zarejestruj się” |
| | 3 | Użytkownik wypełnia formularz rejestracyjny |
| | 4 | Użytkownik otrzymuje potwierdzenie rejestracji |
| Rozszerzenia | Krok | Akcja |
| | 3a | Użytkownik podaje dane, które nie przechodzą walidacji lub są niepełne. <ul style="list-style-type: none">• 3a-1. Użytkownik poprawia wprowadzone dane |
| Wynik | Użytkownik założył konto i jest zalogowany | |

| | | |
|-----------------|-------------------------------|--|
| Scenariusz 2 | Zalogowanie | |
| Opis | Użytkownik chce się zalogować | |
| Aktorzy | Niezalogowany użytkownik | |
| Warunki wstępne | Użytkownik ma konto (S. 1) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę aplikacji |
| | 2 | Użytkownik klika przycisk „Zaloguj się” |
| | 3 | Użytkownik wypełnia formularz logowania |
| | 4 | Użytkownik otrzymuje potwierdzenie logowania |
| Rozszerzenia | Krok | Akcja |
| | 3a | Użytkownik podaje dane, które nie przechodzą walidacji lub są niepełne. <ul style="list-style-type: none"> 3a-1. Użytkownik poprawia wprowadzone dane |
| Wariacje | Krok | Akcja |
| | 3a | Użytkownik może wybrać jedną z kilku opcji logowania: <ul style="list-style-type: none"> Login i hasło Adres e-mail i hasło |
| Wyjątki | Krok | Wyjątek |
| | 3 | Użytkownik nie pamięta swoich danych logowania. |
| Wynik | Użytkownik jest zalogowany | |

| | | |
|-----------------|--|--|
| Scenariusz 3 | Tworzenie fiszki | |
| Opis | Użytkownik chce utworzyć fiszkę | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik jest zalogowany (S. 1 lub S. 2) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik klika przycisk „Dodaj fiszkę” |
| | 3 | Użytkownik wybiera sposób dodania fiszki (do nowej/istniejącej grupy lub standalone) |
| | 4 | Użytkownik wypełnia formularz tworzenia fiszki |
| | 5 | Użytkownik otrzymuje potwierdzenie utworzenia fiszki |
| Wariacje | Krok | Akcja |
| | 3a | Użytkownik dodaje fiszkę do nowej grupy <ul style="list-style-type: none"> 3a-1. Użytkownik tworzy nową grupę (S. 4) |
| | 3b | Użytkownik dodaje fiszkę do istniejącej grupy <ul style="list-style-type: none"> 3b-1. Użytkownik wybiera grupę z listy |
| Wynik | Użytkownik pomyślnie utworzył fiszkę | |

| | | |
|-----------------|--|---|
| Scenariusz 4 | Tworzenie grupy | |
| Opis | Użytkownik chce utworzyć grupę | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik jest zalogowany (S. 1 lub S. 2) | |
| Wyzwalacze | S.3 – tworzenie fiszki | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik klika przycisk „Utwórz grupę” |
| | 3 | Użytkownik wypełnia formularz tworzenia grupy |
| | 4 | Użytkownik otrzymuje potwierdzenie utworzenia grupy |
| Alternatywy | Krok | Akcja |
| | 2a | Użytkownik może także stworzyć grupę w procesie tworzenia fiszki (S. 3) <ul style="list-style-type: none"> • 2a-1. Użytkownik tworzy grupę w trakcie tworzenia fiszki |
| Wariacje | Krok | Akcja |
| | 3a | Użytkownik może tworzyć grupy: <ul style="list-style-type: none"> • Publiczne • Prywatne |
| | 3b | Użytkownik ma wybór czy chce egzekwować na fiszkach w grupie takie atrybuty jak: <ul style="list-style-type: none"> • widoczność (publiczne lub prywatne) • tag (dodanie tagu do wszystkich fiszek) |
| Wynik | Stworzona zostaje grupa. Widoczna jest w sekcji „Moje grupy” | |

| | | |
|-----------------|--|--|
| Scenariusz 5 | Udostępnianie grupy fiszek | |
| Opis | Użytkownik chce udostępnić istniejącą grupę fiszek | |
| Aktorzy | Użytkownik udostępniający (U1), Użytkownik, dla którego grupa jest udostępniana (U2) | |
| Warunki wstępne | Użytkownik stworzył wcześniej grupę | |
| Opis | Krok | Akcja |
| | 1 | U1 wchodzi na stronę główną aplikacji |
| | 2 | U1 wchodzi w swoją grupę w zakładce „Moje grupy” |
| | 3 | U1 wchodzi w stronę grupy i kopiuje link do grupy. |
| | 4 | U2 wchodzi w link do grupy |
| | 5 | U2 jest w stanie przeglądać grupę |
| Rozszerzenia | Krok | Akcja |
| | 4a | Grupa jest prywatna, użytkownik nie ma do niej dostępu <ul style="list-style-type: none"> • 4a-1. U1 zmienia widoczność grupy • 4a-2. U2 ponawia próbę |
| Wariacje | Krok | Akcja |
| | 3a | U1 może udostępnić link na dwa sposoby: <ul style="list-style-type: none"> • kopiuje link z URL przeglądarki • kopiuje link wykorzystując dedykowany przycisk w widoku grupy |
| Wynik | U2 ma dostęp do udostępnionej przez U1 grupy fiszek | |

| | | |
|-----------------|--|---|
| Scenariusz 6 | Wyszukanie grupy po nazwie/tagu | |
| Opis | Użytkownik wyszukuje grupę po tagu lub nazwie. | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik stworzył wcześniej grupę lub istnieje publiczna grupa stworzona przez innego użytkownika (S. 4) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wybiera opcję wyszukiwania na wyszukiwanie grup |
| | 3 | Użytkownik wpisuje wyszukiwaną frazę w pole wyszukiwania |
| | 4 | Z listy wyników wybiera interesującą go grupę |
| Wariacje | Krok | Akcja |
| | 3a | Wpisywana przez użytkownika fraza może być zawarta w: <ul style="list-style-type: none"> • nazwie grupy • tagu |
| | 4a | Wyszukana fraza może być: <ul style="list-style-type: none"> • grupą użytkownika • publiczną grupą stworzoną przez kogoś innego |
| Wyjątki | Krok | Wyjątek |
| | 3a | Nie istnieje grupa, która spełniałaby oczekiwania użytkownika. |
| Wynik | Użytkownik znajduje interesującą go grupę | |

| | | |
|-----------------|--|--|
| Scenariusz 7 | Wyszukiwanie fiszki | |
| Opis | Użytkownik wyszukuje fiszkę po tagu. | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik stworzył wcześniej fiszkę lub istnieje publiczna fiszka stworzona przez innego użytkownika (S. 3) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wybiera opcję wyszukiwania na wyszukiwanie fiszek |
| | 3 | Użytkownik wpisuje wyszukiwaną frazę w pole wyszukiwania |
| | 4 | Z listy wyników wybiera interesującą go fiszkę |
| Wariacje | Krok | Akcja |
| | 4a | Wyszukana fiszka może być: <ul style="list-style-type: none"> fiszka użytkownika publiczną fiszka stworzoną przez kogoś innego |
| Wyjątki | Krok | Wyjątek |
| | 3a | Nie istnieje fiszka, która spełniłaby oczekiwania użytkownika |
| Wynik | Użytkownik znajduje interesującą go fiszkę | |

| | | |
|-----------------|--|---|
| Scenariusz 8 | Edycja fiszki | |
| Opis | Użytkownik edytuje wcześniej dodaną fiszkę | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik dodał wcześniej jakąś fiszkę (S. 3) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wchodzi w sekcję „Moje fiszki” |
| | 3 | Użytkownik klika przycisk edycji przy wybranej fiszce |
| | 4 | Użytkownik wypełnia formularz edycji (tożsamy z formularzem tworzenia) |
| Alternatywy | Krok | Akcja |
| | 2a | <p>Użytkownik może edytować fiszki także z widoku konkretnej grupy fiszek.</p> <ul style="list-style-type: none"> • 2a-1. Użytkownik wchodzi w sekcję „Moje grupy” • 2a-2. Użytkownik wybiera grupę fiszek i wchodzi w listę fiszek |
| Wynik | Użytkownik pomyślnie edytował grupę | |

| | | |
|-----------------|---|---|
| Scenariusz 9 | Edycja grupy | |
| Opis | Użytkownik edytuje wcześniej dodaną grupę | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik dodał wcześniej jakąś grupę (S. 4) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wchodzi w sekcję „Moje grupy” |
| | 3 | Użytkownik klika przycisk edycji grupy |
| | 4 | Użytkownik wypełnia formularz edycji (tożsamy z formularzem tworzenia) |
| Rozszerzenia | Krok | Akcja |
| | 2a | Użytkownik może edytować grupę również z widoku konkretnej grupy 2a-1. Użytkownik wchodzi w stronę konkretnej grupy fiszek |
| Wynik | Użytkownik pomyślnie edytował grupę | |

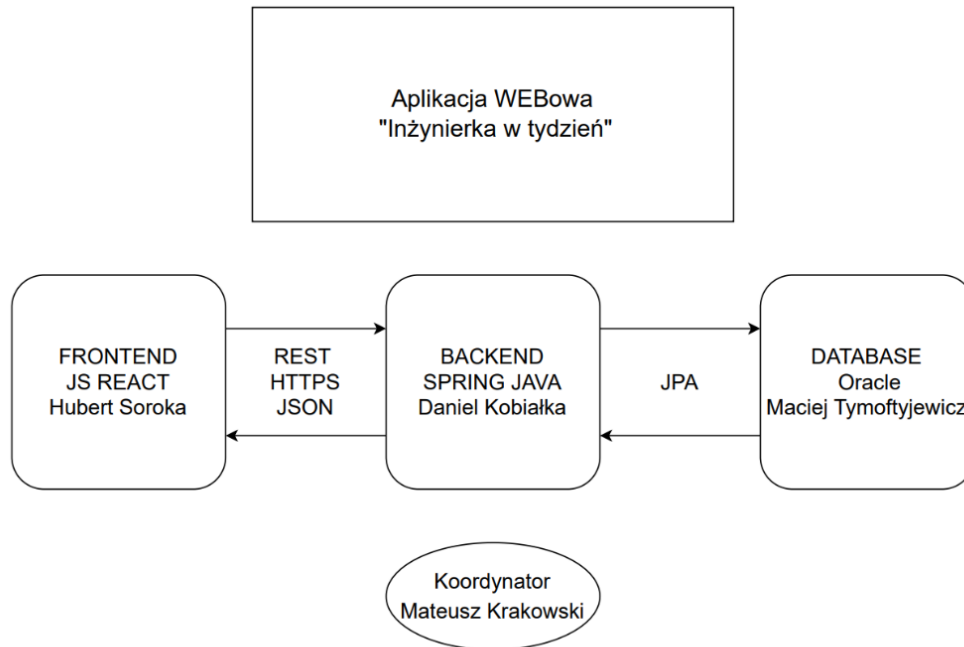
| | | |
|-----------------|--|---|
| Scenariusz 10 | Usunięcie fiszki | |
| Opis | Użytkownik chce usunąć fiszkę | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik dodał wcześniej jakąś fiszkę (S. 3) | |
| Wyzwalacze | S. 11 | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wchodzi w sekcję „Moje fiszki” |
| | 3 | Użytkownik klika przycisk usunięcia przy wybranej fiszce |
| | 4 | Użytkownik potwierdza chęć usunięcia fiszki w pop-upie |
| Alternatywy | Krok | Akcja |
| | 2a | Użytkownik może usunąć fiszki z widoku grupy <ul style="list-style-type: none"> • 2a-1. Użytkownik wchodzi w sekcję „Moje grupy” • 2a-2. Użytkownik wybiera grupę fiszek i wchodzi w listę fiszek |
| | 2+ | Użytkownik może usunąć fiszkę przy okazji usuwania grupy fiszek (S. 11) |
| Wynik | Użytkownik usunął fiszkę | |

| | | |
|-----------------|---|--|
| Scenariusz 11 | Usunięcie grupy | |
| Opis | Użytkownik chce usunąć grupę | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik dodał wcześniej jakąś grupę (S. 4) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wchodzi w sekcję „Moje grupy” |
| | 3 | Użytkownik klika przycisk usunięcia przy wybranej grupie |
| | 4 | Użytkownik potwierdza chęć usunięcia grupy w pop-upie |
| Rozszerzenia | Krok | Akcja |
| | 2a | Użytkownik może usunąć grupę również widoku konkretnej grupy 2a-1. Użytkownik wchodzi w stronę konkretnej grupy fiszek |
| Wariacje | Krok | Akcja |
| | 4a | Użytkownik usuwając grupę może: <ul style="list-style-type: none"> • Usunąć wszystkie powiązane fiszki • Pozostawić wszystkie fiszki jako standalone |
| Wynik | Użytkownik pomyślnie usuwa grupę | |

| | | |
|-----------------|--|---|
| Scenariusz 12 | Dodanie fiszki do grupy | |
| Opis | Użytkownik chce dodać fiszkę do grupy | |
| Aktorzy | Użytkownik | |
| Warunki wstępne | Użytkownik stworzył wcześniej grupę oraz fiszkę (S. 3, S. 4) | |
| Opis | Krok | Akcja |
| | 1 | Użytkownik wchodzi na stronę główną aplikacji |
| | 2 | Użytkownik wchodzi w sekcję „Moje grupy” |
| | 3 | Użytkownik wchodzi w stronę konkretnej grupy fiszek |
| | 4 | Użytkownik używa przycisku dodania nowej fiszki |
| | 5 | Użytkownik wypełnia formularz tworzenia fiszki |
| Alternatywy | Krok | Akcja |
| | 2+ | <p>Użytkownik może dodać istniejącą fiszkę do grupy również z położenia listy fiszek</p> <ul style="list-style-type: none"> • 2a. Użytkownik wchodzi w sekcję „Moje fiszki” • 3a. Użytkownik używa przycisku dodawania do grupy • 4a. Użytkownik wybiera grupę z listy |
| | 4a | <p>Wyszukana fiszka może być:</p> <ul style="list-style-type: none"> • grupą użytkownika • publiczną grupą stworzoną przez kogoś innego |
| Wynik | Użytkownik pomyślnie dodaje fiszkę do grupy | |

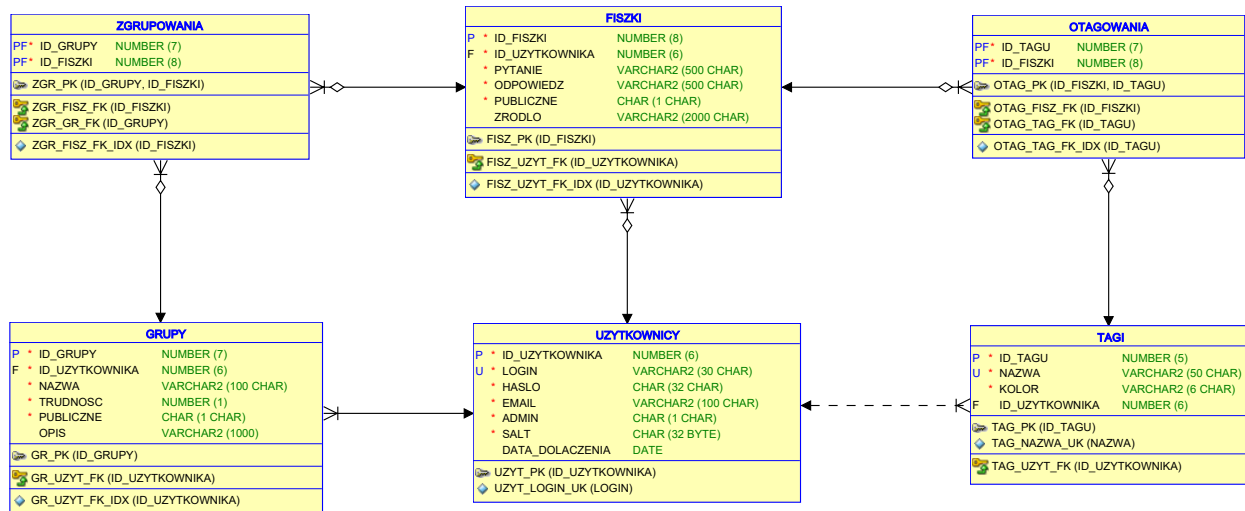
4 Specyfikacja projektowa

4.1 Metoda Realizacji



Projekt wykonywany z użyciem bazy danych Oracle (udostępnianej przez uczelnię), backendu w Javie z użyciem Spring oraz niezależnego frontendu w typescriptie z Reactem. Widać jasno trzy rozłączne części, każda z nich funkcjonująca niezależnie od pozostałych, każda uruchamiania oddzielnie. Baza danych komunikuje się z backendem poprzez interfejs Spring Data JPA, który wykorzystuje driver JDBC. Backend wystawia Restowe API, przez które frontend nawiązuje połączenie. Dane przesyłane Jsonem, protokołem http. W celach poglądowych, do dokumentacji załączamy diagram ERD i Logiczny bazy danych.

4.2 Logiczny model danych



Model również znajduje się w załączniku Logical.pdf w lepszej rozdzielczości (Link na końcu tego pliku).

Programowanie w bazie danych ograniczyliśmy do minimum. Wykożystaliśmy:

Więzy CHECK UZYT_ADMIN_CHECK, GR_TRUDNOSC_CHECK i FIS_PUBLICZNE_CHECK sprawdzające, czy dane atrybuty są z danego zbioru dyskretnego wartości, np. TRUDNOSC IN (1, 2, 3)

TRIGGERY używamy do:

- realizowania ograniczeń nietransferowalności
- nadawania identyfikatorów z sekwencji dla nowo zapisanych danych

Nałożyliśmy dodatkowe indeksy na klucze obce.

Użyliśmy UK (np. TAG_NAZWA_UK) aby zachować unikalność nazw tagów.

4.3 Model obiektowy systemu

Poniżej opisywany Diagram przedstawiający architekturę backendu znajdują się w pliku PZSP2-UML-2_ciemny.svg i PZSP2-UML-2_jasny.svg (Link do załączników na końcu tego pliku).

UI oraz ZAPYTANIA są realizowane po stronie aplikacji Reactowej. Nie stworzyliśmy diagramu dla komponentów frontendowych, ponieważ z punktu widzenia organizacji architektury, całość danych, które przetwarzają, będzie przechodzić przez klasę Requests z diagramu. Klasy wyżej niż ZAPYTANIA odpowiadają tylko prezentacji danych, a nie ich przetwarzaniu.

Klasy z grupy MODELE są zamieszczone na boku warstw SERWISÓW oraz REPOZYTO-

RIÓW, ponieważ są często wykorzystywane przez obie z nich. Nie zamieszczono linii notacji UML przy każdym wykorzystaniu tych klas, ponieważ uczyniłoby to diagram nieczytelny. Te klasy są bezpośrednio mapowane na tabele relacyjne w bazie danych, reprezentują wszystkie główne tabele, pozostałe służą tylko reprezentacji związków N-N bez atrybutów pomiędzy nimi.

Istnieją też klasy pomocnicze wykorzystywane do obsługi api, w katalogu models/api. Unikamy wysyłania klas modeli mapowanych do bazy danych na frontend, aby uniknąć przesyłania nadmiarowych danych, oraz nie udostępniać pól wszystkich klas powiązanych (np. nie chcemy udostępniać maila twórcy każdej fiszki przy pobieraniu ich listy).

Kiedy niemożliwe lub zabronione jest dalsze wykonanie zapytania, rzucany jest wyjątek typu `BackendException`. Te wyjątki są obsługiwane przez `ApiExceptionHandler`, który wysyła na frontend komunikat o błędzie z kodem 418 (stosowany na potrzeby testów, przy rozszerzeniu obsługi wyjątków będzie zmieniony).

Na frontendzie w pliku `Requests.ts` jest zaimplementowane generyczne obsługiwanie zapytań, dzięki czemu niezależnie od typu danych, w kontrolkach reactowych można bardzo podobnie pobierać dane z backendu.

Nasz pomysł na architekturę najprościej można podsumować następująco:

1. Komponent frontendowy, potrzebując konkretnych danych, wywołuje jedną z funkcji z klasy statycznej `Requests`
2. Każda funkcja tej klasy odpowiada jednej akcji w jednym z kontrolerów
3. Kontrolery wywołują metodę z odpowiedniego serwisu, który jest polem kontrolera. Kontrolery zawierają minimum logiki „biznesowej” programu. Odsyłają dane w klasie z sufiksem `Response`, są to proste klasy bez metod, które zawierają tylko to, co będzie potrzebne na frontendzie w kontekście danej metody.
4. Serwis wykonuje operacje na obiektach z modelu dostarczonych przez repozytoria tak, aby móc zwrócić wymagane dane lub rzucić wyjątek, który jest obsługiwany przez `ApiExceptionHandler` i generowana jest odpowiedź zawierająca komunikat wyjątku.
5. Repozytoria dziedziczą po `JpaRepository`, dostarczonym przez Spring. Dzięki czemu nie trzeba ich implementować, wystarczy stworzyć sam interfejs, nazywając metody zgodnie ze standardem Spring. Zapytania im odpowiadające wygenerują się automatycznie. W przypadku, gdy potrzeba wykonać trudniejsze zapytanie, wystarczy dodać adnotację `@Query(...)` i podać zapytanie SQLowe.
6. Spring wykonuje zapytania i zwraca obiekty z modelu przedstawiające relacyjne dane z bazy danych.

4.4 Intefrejs użytkownika

SCHEMATY UI

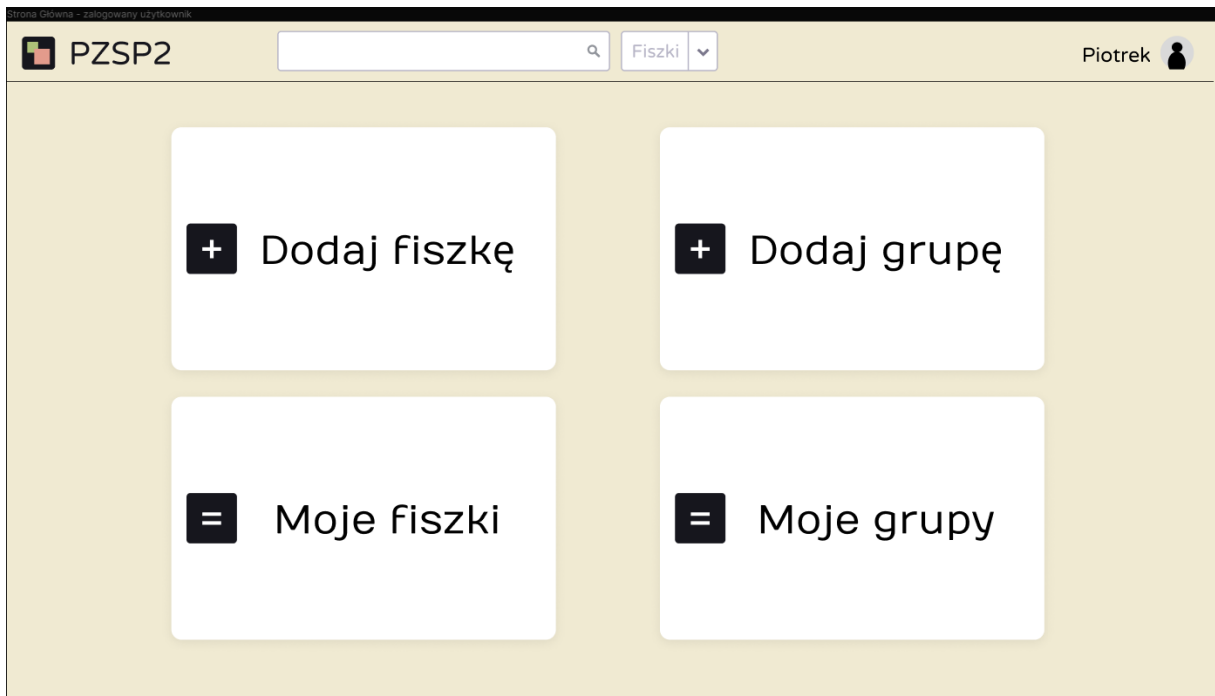
Schematy stworzone przy pomocy narzędzia Figma:

<https://www.figma.com/file/egM5q8moynTEK0gTQx2oQw/PZSP2-UI>

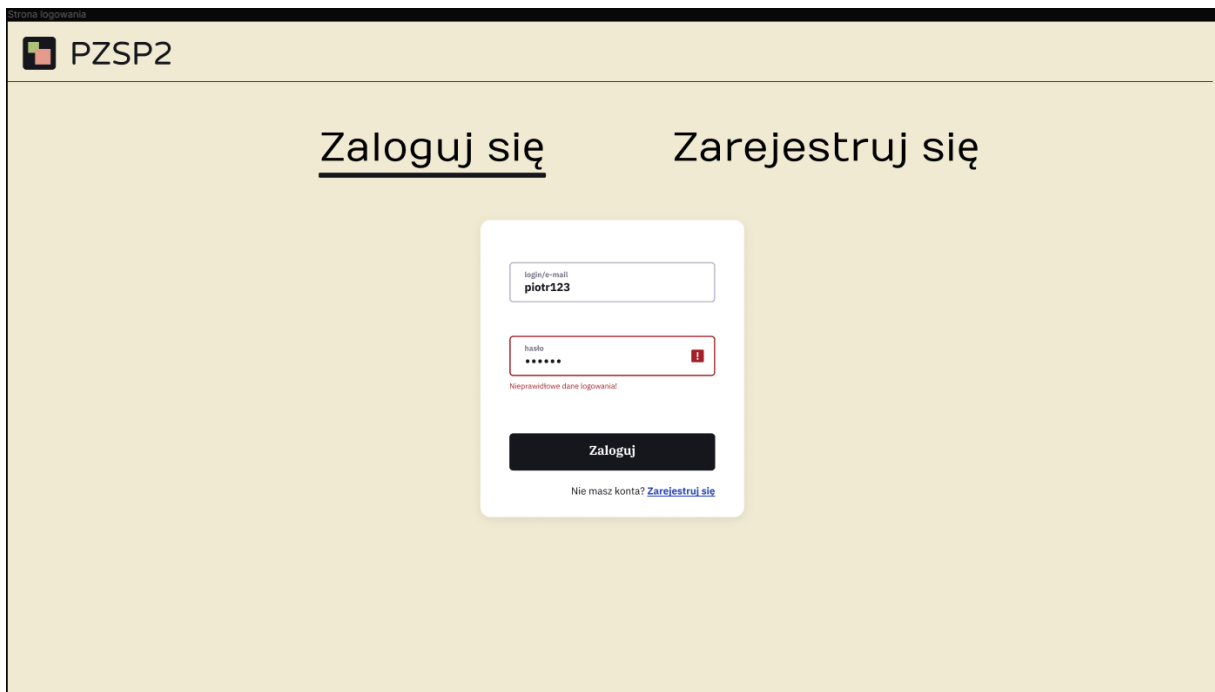
Prototyp (mock-up) UI:

<https://www.figma.com/proto/egM5q8moynTEK0gTQx2oQw/PZSP2-UI...>

STRONA GŁÓWNA



STRONA LOGOWANIA



STRONA REJESTRACJI

Strona rejestracji

 PZSP2

Zaloguj się Zarejestruj się

login
piotr123

e-mail
piotr.kowalski.stud@pw.edu.pl

hasło


powtórz hasło


Zarejestruj

Masz już konto? [Zaloguj się](#)

DODAJ/EDYTUJ FISZKĘ

Dodaj/edytuj fiszkę

 PZSP2

Piotrek 

Pytanie

Jakie przydatne informacje możemy uzyskać wykonując test balbinka?

Odpowiedź

Można dowiedzieć się, czy jest się lokomotywą, czy może jednak lidarem.

Źródło

-Ćwiczenia z przedmiotu PZSP2 (moodle)
-https://pl.wikipedia.org/wiki/Role_w_zespole

Grupy: PZSP2 ✕

Tagi: PZSP2 ✕ INFORMATYKA ✕

☒ Publiczna

DODAJ/EDYTUJ GRUPĘ

Moja grupa

PZSP2

Fiszki

Piotrek

nazwa grupy
AISDI W1-5

opis
Notatki z wykładów od 1 do 5 z przedmiotu AISDI

1 Trudność

✓ Publiczna

Dodaj grupę

MOJE FISZKI

Moje fiszki

PZSP2

Fiszki

Piotrek

Filtruj po tagach...

Tagi: AISDI x INFORMATYKA x

Ucz się Test

| Pytanie | Grupy | Tagi | Widoczność | |
|--|-----------------------------|-------------------------|------------|---|
| Opisz rodzaje hypervisora VM | <brak> | INFORMATYKA | Prywatna | > |
| Przekształcanie gramatyk bezkontekstowych... | AISDI W1-5, AISDI ALGORYTMY | AISDI | Publiczna | > |
| Opisz kolejne kroki operacji wstawiania w drz... | AISDI W1-5, ALGORYTMY | AISDI, INFORMATYKA, AVL | Publiczna | > |
| | | | | |

< 1 2 3 >

MOJE GRUPY

Moje grupy

PZSP2

| Nazwa | Opis | Trudność | Widoczność | Szczegóły |
|-------------|--|----------|------------|-----------|
| AISDI | Pytania z przedmiotu AISDI | 5/10 | Prywatna | > |
| AISDI W1-5 | Wykłady 1-5 z przedmiotu AISDI | 7/10 | Publiczna | > |
| INFORMATYKA | Wszelkiego rodzaju pytania z dziedziny inform... | 2/10 | Publiczna | > |
| | | | | |

< 1 2 3 >

STRONA GRUPY

Grupa w grupie

PZSP2

AISDI

Właściciel: Ty

Opis

Wykłady z przedmiotu AISDI

| Pytanie | Odpowiedź | Źródło | Tagi | |
|--|--|---|--------------------|---|
| Gdzie tupta noca jeż? | Do lasu narobić hałasu | https://odpowiedziny pytania.pl/dokad-noca-tupta... | ZWIERZĘTA | > |
| Przekształcania gramatyk bezkontekstowych... | Każdą gramatykę bezkontekstową G = <N, T, P, Z>... | Wykłady AISDI, http://carne.plid-linux.org/~evi | AISDI, ALGORYTMY | > |
| Opisz kolejne kroki operacji wstawiania w drz... | Podczas operacji insert tak samo jak dla zwykły... | AISDI, http://smurf.mimuw.edu.pl/node/341 | AISDI, INFORMATYKA | > |
| | | | | |

< 1 2 3 >

STRONA FISZKI

Strona fiszki

PZSP2

Fiszki

Piotrek

Właściciel: Ty

Pytanie

Opisz kolejne kroki operacji wstawiania w drzewie AVL

Odpowiedź

Podczas operacji Insert tak samo jak dla zwykłych drzew BST schodzimy po ścieżce od korzenia w dół do węzła zewnętrznego NULL i w jego miejscu tworzymy nowy liść ze wstawianym kluczem. Następnie wracamy po ścieżce do korzenia, aktualizując współczynniki zrównowazenia. Jeśli stwierdzamy, że wysokość aktualnie rozealanego poddrzewa nie zmieniła się w stosunku do sytuacji przed wykonaniem insert, to kończymy operację a jeśli stwierdzamy, że wysokość poddrzewa wzrosła (niegdyj więcej niż o 1), kontynuujemy mniej w górę drzewa. Jeśli w wyniku wzrostu wysokości jednego z poddrzew aktualnie rozealanego węzła został w nim zaburzony warunek AVL, to przywracamy go za pomocą rotacji. Z dokładnością do symetrii mamy wtedy dwa przypadki (w obydwu zakładamy, że op jest węzłem, w którym zaburzony został warunek AVL, wskutek wzrostu wysokości jego prawego poddrzewa o korzeniu op; w pierwszym przypadku zakładamy, że wzrosła wysokość prawego poddrzewa op, a w drugim – lewego).

Źródło

<http://smurf.mimuw.edu.pl/node/341>

Grupy: AISDI, ALGORYTMY

Tagi: AISDI

Edytuj fiszkę

WYSZUKIWANIE FISZEK

Wyszukiwanie fiszek

PZSP2

Fiszki

Piotrek

Wyszukiwana fraza: "AISDI"

Typ wyników: Fiszki

Ucz się

Test

| Pytanie | Grupy | Tagi | Właściciel | |
|--|-----------------------------|-------------------------|-------------|---|
| Przekształcania gramatyk bezkontekstowych... | AISDI W1-5, AISDI ALGORYTMY | AISDI | janek1 | > |
| Opisz kolejne kroki operacji wstawiania w drz... | DRZEWA | AISDI, INFORMATYKA, AVL | mariusz2000 | > |
| | | | | |

<

1

2

3

>

WYSZUKIWANIE GRUP

Wyszukiwana grupa

PZSP2

Fiszki

Piotrek

Wyszukiwana fraza: "AISDI"

Typ wyników: Grupy

| Nazwa | Opis | Trudność | Właściciel | Szczegóły |
|------------|--|----------|------------|-----------|
| AISDI | Pytania z przedmiotu AISDI | 5/10 | janek1 | > |
| AISDI W1-5 | Wykłady 1-5 z przedmiotu AISDI | 7/10 | janek1 | > |
| DRZEWA | Wszystkie pytanie o drzewach AVL, BST itd. | 2/10 | mariusz000 | > |
| | | | | |

<

1

2

3

>

STRONA SCENARIUSZA TESTOWEGO

Test

PZSP2

Fiszki

Piotrek

1/2

Pytanie

100%

Opisz kolejne kroki operacji wstawiania w drzewie AVL

Nie umiem

Umieć

PRZEGLĄDANIE FISZEK

Przeglądanie fiszek 1

PZSP2

Fiszki

Piotrek

1/2

Pytanie

Opisz kolejne kroki operacji wstawiania w drzewie AVL

Odpowiedź

Podczas operacji Insert tak samo jak dla zwykłych drzew BST schodzimy po ścieżce od korzenia w dół do węzła zewnętrznego NULL i w jego miejscu tworzymy nowy śluz ze wstawianym kluczem. Następnie wracamy po ścieżce do korzenia, aktualizując współczynniki zrównowazenia. Jeśli stwierdzamy, że wysokość aktualnie rosnącego poddrzewa nie zmieniła się w stosunku do sytuacji przed wykonaniem Insert, to kończymy operację, a jeśli stwierdzamy, że wysokość poddrzewa wzrosła (mogła wzrosnąć co najwyżej o 1), kontynuujemy marsz w górę drzewa. Jeśli w wyniku wzrostu wysokości jednego z poddrzew aktualnie rosnącego węzła został w nim zaburzony warunek AVL, to przywracamy go za pomocą rotacji. Z dokładnością do symetrii mamy wtedy dwa przypadki (w obydwu zakładamy, że op jest węzłem, w którym zaburzony został warunek AVL, wskutek wzrostu wysokości jego prawego poddrzewa o korzeniu qq, w pierwszym przypadku zakładamy, że wzrosła wysokość prawego poddrzewa qq, a w drugim – lewego).

Źródło

<http://smurf.mimuw.edu.pl/node/341>

Poprzednia fiszka

Następna fiszka

PROFIL UŻYTKOWNIKA

Profil użytkownika

PZSP2

Fiszki

Piotrek

Piotr

E-mail: piotr.nowak@pw.edu.pl

Data dołączenia: 12-10-2022

Liczba fiszek: 12

Liczba grup: 4

Zmień nazwę użytkownika

Zmień adres e-mail

Zmień hasło

Usuń konto

4.5 Standardy obsługi błędów i sytuacji wyjątkowych

Aby zachować standardy obsługi błędów i sytuacji wyjątkowych należy:

- Sporządzać dziennik błędów, rejestrowanie błędów i wyjątków, które wystąpiły w systemie, pozwala to na łatwiejsze ich rozwiązanie.
- Sporządzać backupy danych systemu oraz przechowywać je w bezpiecznym miejscu.
- Monitorować system, pozwala to na szybkie zauważenie i rozwiązanie ewentualnych problemów.
- Informować użytkowników o wystąpieniu błędu i jego przyczynie.
- Skontaktować się z dostawcą oprogramowania wraz z kodem błędu z wcześniej wspomnianego dziennika błędów.

5 Specyfikacja testów

5.1 Scenariusze testów funkcjonalnych

Scenariusze testów funkcjonalnych z których korzystaliśmy w naszym projekcie:

1. Logowanie: Sprawdzenie czy użytkownik jest w stanie zalogować się do systemu przy użyciu poprawnej nazwy użytkownika i hasła.
2. Przeglądanie fiszek i grup: Sprawdzenie czy użytkownik jest w stanie przeglądać fiszki i grupy przechowywane w bazie danych.
3. Dodawanie fiszek i grup: Sprawdzenie czy użytkownik jest w stanie dodać nowe fiszki i grupy do systemu poprzez wypełnienie formularza i kliknięcie przycisku "Dodaj".
4. Edycja fiszek i grup: Sprawdzenie czy użytkownik jest w stanie edytować już istniejące fiszki i grupy poprzez kliknięcie przycisku "Edytuj".
5. Usuwanie fiszek i grup: Sprawdzenie czy użytkownik jest w stanie usunąć fiszki i grupy z systemu poprzez kliknięcie przycisku "Usuń".
6. Wylogowanie: Sprawdzenie czy użytkownik jest w stanie wylogować się z systemu poprzez kliknięcie przycisku "Wyloguj się".
7. Backup: Sprawdzenie czy system jest w stanie utworzyć i przywrócić kopie zapasowe.
8. Testowanie różnych przeglądarek: Sprawdzenie czy system działa poprawnie na różnych przeglądarkach.
9. Testowanie na różnych urządzeniach: Sprawdzenie czy system działa poprawnie na różnych urządzeniach, takich jak komputery stacjonarne, laptopy, tablety i telefony komórkowe.
10. Testowanie różnych rozdzielczości ekranów: Sprawdzenie czy system działa poprawnie na różnych rozdzielczościach ekranów.

6 Podręcznik użytkownika

6.1 Operacje możliwe do wykonania w aplikacji

1. Założenie konta użytkownika: Aby założyć konto użytkownika należy kliknąć przycisk "Załącz konto", uzupełnić formularz i potwierdzić rejestrację.
2. Zalogowanie się na konto: Aby zalogować się na swoje konto, należy wprowadzić swoją nazwę użytkownika oraz hasło i kliknąć przycisk "Zaloguj się".
3. Wylogowanie się z konta: Aby wylogować się z konta, należy kliknąć przycisk "Wyloguj się" znajdujący się w prawym górnym rogu aplikacji.
4. Tworzenie, edytowanie oraz usuwanie fiszek: Aby stworzyć nową fiszkę, należy kliknąć przycisk "Dodaj fiszkę" i uzupełnić formularz. Aby edytować lub usunąć fiszkę, należy przejść do "Moje fiszki", wybrać fiszkę klikając na przycisk (>) po prawej stronie fiszki, a następnie kliknąć przycisk požądanej przez nas opcji, przyciski znajdują się na dole ekranu.
5. Tworzenie, edytowanie oraz usuwanie grup: Aby stworzyć nową grupę, należy kliknąć przycisk "Stwórz nową grupę" i uzupełnić formularz. Aby edytować lub usunąć grupę, należy przejść do "Moje grupy", wybrać grupę którą chcemy edytować, a następnie kliknąć przycisk požądanej przez nas opcji, przyciski znajdują się na dole ekranu. TODO HERE
6. Dodawanie tagów: Aby dodać tagi do fiszki lub grupy, TODO HERE
7. Wyszukanie fiszek i grup po tagu: Aby wyszukać fiszki lub grupy po tagu, należy wykorzystać wyszukiwarkę znajdującą się nad tabelką na stronie "Moje fiszki" lub "Publiczne fiszki". Wyszukiwarka znajduje się w górnej części ekranu.
8. Dodanie fiszki do grupy: Operacje tą można wykonać zarówno z poziomu samej fiszki jak i grupy. Można to zrobić przy dodawaniu oraz edytowaniu grupy, wystarczy zaznaczyć okienko po lewej stronie fiszki w ekranie "Dodaj grupę" lub podczas edycji grupy. Fiszke do grupy można dodać przy samym wykonywaniu operacji dodawania fiszki, wystarczy wybrać grupy docelowe w przycisku "Wybierz grupy". To samo można uczynić podczas edytowania fiszki.
9. Przechodzenie przez tryb nauki dla danej grupy: Aby rozpocząć tryb nauki dla danej grupy, należy przejść z menu do "Moje grupy" lub "Publiczne grupy", wybrać grupę klikając przycisk (>) po prawej stronie, a następnie kliknąć przycisk "Ucz się".
10. Przechodzenie przez tryb testu dla danej grupy: Aby rozpocząć testu dla danej grupy, należy przejść z menu do "Moje grupy" lub "Publiczne grupy", wybrać grupę klikając przycisk (>) po prawej stronie, a następnie kliknąć przycisk "Test".

7 Podręcznik administratora

7.1 Instrukcja instalacji i konfiguracji systemu

1. Pobierz i zainstaluj potrzebne narzędzia: Node.js, Java Development Kit (JDK) 17, Apache Maven.
2. Sklonuj kod źródłowy projektu z repozytorium na swój komputer.
3. Skonfiguruj połączenie między aplikacją backendową a bazą danych w pliku `application.properties`. Należy podać adres bazy, hasło i nazwę użytkownika.
4. Przejdź do katalogu `/api` i uruchom polecenie `"mvn clean install"` w celu skompilowania i zainstalowania aplikacji.
5. Uruchom polecenie `"mvn spring-boot:run"` aby uruchomić aplikację backendową.
6. Przejdź do katalogu `/platform` i uruchom polecenie `"npm install"` w celu zainstalowania potrzebnych zależności.
7. Uruchom polecenie `"npm start"` w celu uruchomienia aplikacji frontendowej.
8. Uruchom aplikację i sprawdź, czy wszystkie funkcjonalności działają poprawnie, w razie problemów zgłoś je do dostawcy oprogramowania.

7.2 Instrukcja aktualizacji oprogramowania

1. Przed rozpoczęciem aktualizacji zalecane jest wykonanie kopii zapasowej bazy danych.
2. Pobierz nowy kod źródłowy z repozytorium.
3. Przejdź do katalogu `/api` i uruchom polecenie `"mvn clean install"` w celu zaktualizowania i zainstalowania aplikacji.
4. Uruchom polecenie `"mvn spring-boot:run"` aby uruchomić zaktualizowaną wersję aplikacji backendowej.
5. Przejdź do katalogu `/platform` i uruchom polecenie `"npm install"` w celu zaktualizowania zależności.
6. Uruchom polecenie `"npm run build"` aby przebudować aplikację frontendową.
7. Sprawdź czy aplikacja działa poprawnie po aktualizacji.
8. Jeśli wystąpiły jakiegokolwiek błędy, skontaktuj się z dostawcą aplikacji.

7.3 Instrukcja tworzenia kopii zapasowej

1. Zaloguj się do serwera bazy danych Oracle.
2. Uruchom polecenie "exp" by zaimportować plik kopii zapasowej bazy danych. Upewnij się, że plik kopii zapasowej zostanie zapisany w bezpiecznym miejscu.
3. Skopiuj katalog z kodem źródłowym aplikacji oraz pliki konfiguracyjne do bezpiecznego miejsca.
4. Sprawdź, czy kopia zapasowa jest kompletna i możliwa do odtworzenia.

7.4 Instrukcja odtwarzania systemu

1. Zaloguj się do serwera bazy danych Oracle.
2. Uruchom polecenie "imp" aby zaimportować plik kopii zapasowej bazy danych.
3. Skopiuj katalog z kodem źródłowym aplikacji oraz pliki konfiguracyjne z kopii zapasowej na serwer.
4. Przejdź do katalogu z kodem źródłowym i uruchom polecenie "mvn clean install" w celu zainstalowania aplikacji.
5. Uruchom polecenie "mvn spring-boot:run" aby uruchomić aplikację.
6. Sprawdź czy aplikacja działa poprawnie.

8 Przydatne linki

8.1 Link do repozytorium projektu

LINK DO REPOZYTORIUM PROJEKTU

8.2 Link do załączników

LINK DO ZAŁĄCZNIKÓW PROJEKTU