

## Wprowadzenie do testów

Testowanie projektu ma na celu sprawdzić:

1. Spełnienie wymagań funkcjonalnych.
2. Spełnienie wymagań niefunkcjonalnych.
3. Czy istnieją elementy, które mogą skutkować nieoczekiwany wynik.
4. Czy projekt jest odporny na dane poza założeniami.
5. Czy projekt nie zawiera ukrytych błędów.

## Plan testów

Proces testowania aplikacji będzie podzielony na kilka etapów. Rozpoczniemy od testów jednostkowych, które skupią się na przetestowaniu najmniejszych części kodu. Następnie skoncentrujemy się na testach systemowych, które będą kluczowe i najważniejsze, ponieważ sprawdzą one funkcjonowanie systemu jako całości oraz pokryją jak najwięcej funkcji. Testy systemowe obejmą wszystkie warunki testowe i uwzględnią różne scenariusze użycia.

## Narzędzia do testowania

Głównymi narzędziami są *JUnit*, *SpringTest*, *Postman*, które posiadają różne narzędzia do testowania aplikacji webowych oraz bazy danych.

## Testy

Głównym celem testów jest sprawdzenie całej aplikacji ze strony użytkownika oraz rozpatrzyć nieoczekiwane przypadki użycia.

1. Sprawdzenie poprawności działania nawigacji pomiędzy oknami w aplikacji
  - a. Czy kliknięcie odpowiednią kartę powoduje wyświetlenie właściwej zawartości.
  - b. Czy odpowiednie okna otwierają się i zamykają w odpowiednich momentach.
  - c. Czy każdy przycisk działa lub nie działa.
  - d. Czy listy wyświetlają odpowiednią zawartość.
2. Testowanie kontrolerów
  - a. Poprawność dodawania danych.
  - b. Poprawność wyświetlenia danych.
  - c. Poprawność dodawania uzupełnień.
  - d. Poprawność zachowania danych.
3. Testy integracyjne
  - a. Testowanie współdziałania różnych funkcji.
  - b. Sprawdzenie możliwych defektów pomiędzy funkcjami.
  - c. Sprawdzenie dużej liczby przypadków użycia.

4. Testowanie wymagań niefunkcyjnych
  - a. Sprawdzenie otwartości na różnych urządzeniach.
  - b. Symulacja awarii oraz wykorzystanie kopii zapasowych do odtwarzania danych.
  - c. Zwiększenie liczby obiektów w aplikacji dla testowania skalowalności.
  - d. Sprawdzenie filtracji zapytań.
5. Testowanie logiki biznesowej
  - a. Sprawdzenie modelu.
  - b. Sprawdzenie poprawności danych.

## Testy Jednostkowe

### Controller:

- AlbumControllerTest.java
- GreetingControllerTest.java
- CreateLessonControllerTest.java
- UserControllerTest.java
- LessonControllerTest.java

### Model:

- LessonInfoTest.java
- UserTest.java
- AlbumTest.java
- LessonTest.java

### Security:

- AuthenticationTest.java
- SecurityConfigTest.java

## Odnalezione błędy

- 1) LessonDao method findLessonById zwraca null gdy test nie istnieje w bazie co powoduje błąd HTTP 500.
- 2) Zapytanie /searchLesson dziedziczy błąd 1.
- 3) AlbumDao method getAlbumByTypeLevel zwraca null gdy album nie istnieje w bazie co powoduje błąd HTTP 500.
- 4) Video player nie może otworzyć URL z pustym symbolem na początku.
- 5) Testy stworzone przez administratora nie są dostępne innym użytkownikom.
- 6) Algorytm sprawdzający odpowiedzi użytkownika wyświetla tylko pierwszy symbol odpowiedzi zamiast całego słowa.
- 7) Możliwość wpisać puste imię użytkownika podczas logowania.
- 8) Nie sprawdzenie danych podczas stworzenia nowych testów:
  - a. Nie sprawdzenie poprawności URL.
  - b. Nie sprawdzenie poprawności pytań oraz odpowiedzi, możliwość wpisać puste pytanie oraz odpowiedź.
- 9) Niepoprawny rozmiar opisu, gdy test jest jeden w albumie.

## Pokrycie testowe

### Controller:

- Class 100%
- Method 100%
- Lines 100%

### Model:

- Class 100%
- Method 98%
- Lines 98%

### Security:

- Class 100%
- Method 100%
- Lines 100%

### Database:

- Class 50%
- Method 33%
- Lines 18%

## Testowanie bazy oparte na use-case

Baza współdziała z serwerem za pomocą ORM za pomocą Hibernate dlatego testowanie bazy było zrobione na wysokim poziomie dla sprawdzenia poprawności zachowania oraz przetwarzania danych.

## Końcowy raport

Wszystkie zaimplementowane testy skutecznie sprawdziły poprawność działania aplikacji. Dzięki nim mamy pewność w spełnieniu wymagań projektowych. Testy systemowe oraz automatyczne miały duże znaczenie podczas tworzenia aplikacji i znacznie skróciły czas lokalizacji błędów. Znaleziona liczba błędów jest oczekiwana co wskazuje na duży poziom ufności aplikacji. Większość testów została zaimplementowana zgodnie z planem.