



## **Laboratorium 8**

### **Jakarta Server Faces + Jakarta Persistence + Bean Validation**

Studenci w ramach zajęć laboratoryjnych zapoznają się z zaawansowanymi elementami znanych już specyfikacji Jakarta Server Faces i Jakarta Persistence oraz podstawami specyfikacji Bean Validation.

Należy zrealizować następujące zadania:

1. Kontrola wersji dla klasy encyjnej elementu. Dla klasy encyjnej elementu należy skonfigurować mechanizm optimistic locking. Numer wersji powinien być wyświetlany na liście elementów. Mechanizm optimistic locking musi być poprawnie obsługiwany po stronie widoku. Oznacza to, że w przypadku wykrycia próby zaktualizowania niepoprawnej wersji użytkownik powinien być o tym powiadomiony. Dodatkowo powinien mieć dostęp zarówno do aktualnego stanu z bazy jak i danych, które poprzednio wprowadził w formularzu edycji, a których nie udało się zapisać do bazy. (0.5 + 0.5 pkt)
2. Data stworzenia i modyfikacji dla klasy encyjnej elementu. Dla klasy encyjnej elementu należy skonfigurować mechanizm automatycznej aktualizacji daty stworzenia i ostatniej aktualizacji obiektu w bazie danych. Data stworzenia i data ostatniej aktualizacji powinny być wyświetlane na liście elementów. (0.5 + 0.5 pkt)
3. Zapytania Criteria API. Wszędzie gdzie są używane zapytania JPQL powinny być one zamienione na zapytania Criteria API. (0.5 + 0.5 pkt)
4. Filtrowanie listy elementów. Należy zapewnić możliwość filtrowania listy elementów po wszystkich polach łączonych operatorem AND. To znaczy, że jeśli klasa ma dwa pola (pomijamy id generowane automatycznie) to użytkownik może wprowadzić dwie wartości i każda z nich będzie musiała być spełniona. Wszystkie wartości są opcjonalne. Oznacza to, że jeśli użytkownik nie wprowadzi jakiejś wartości to nie będzie ona brana pod uwagę. Niewprowadzenie jakiegokolwiek wartości skutkuje wyświetleniem wszystkich dostępnych elementów. Zapytanie musi być budowane dynamicznie za pomocą Criteria API. (1 + 1 pkt)
5. Walidacja elementów. Należy skorzystać z gotowych adnotacji dostępnych w Bean Validation do zwalidowania obiektu elementu podczas tworzenia. Należy też zaproponować i zaimplementować własny walidator dla jednego z pól. Wiadomości błędów należy wyświetlić na stronie edycji. (0.5 + 0.5 pkt)

Poprawne wykonanie zadań wymaga m. in. właściwej dekompozycji na komponenty, poprawnego wykorzystania tagów HTML (w tym tagów semantycznych) i stylów CSS, poprawnego zarządzania stanem obiektów zarządzanych oraz poprawnego wykorzystania modelu JPA w budowaniu zapytań.

**Uwaga:** Podczas sprawdzania zadania będą sprawdzane tylko wywołania w ramach widoków JSF.



**POLITECHNIKA  
GDAŃSKA**

**Narzędzia i Aplikacje  
Jakarta EE**

**Uwaga:** Wszystkie rozwiązania należy zachować na potrzeby kolejnych zadań, chyba, że w kolejnych instrukcjach będzie napisane inaczej.