

## Programowanie zaawansowanych aplikacji webowych

Projekt: regulamin kursu i zasady oceniania

Data: 02.10.2019

Autor: Marcin Jodłowiec

1. Podstawowym celem zajęć jest:

- a. zaprojektowanie (P),
- b. implementacja (I),
- c. przebadanie (R) oraz
- d. optymalizacja (O)

**aplikacji webowej.**

2. Aby zrealizować cel, studenci mogą pracować w 2 lub 3-osobowych grupach.

3. Na zajęciach każda z grup zobowiązana jest przedstawić i **zreferować** prowadzącemu pisemny raport dotyczący wykonanych działań, który odzwierciedlać będzie wszelkie elementy dokumentacyjne oraz/lub opisywać wykonane artefakty cyfrowe. Każdy z członków grupy musi być przygotowany merytorycznie do rozmowy z prowadzącym w zakresie postępów w projekcie.

4. Na zajęciach od 2 do 14 studenci zdobywają punkty, które przekładają się na ocenę końcową z projektu. Liczba uzyskanych punktów będzie proporcjonalna do zaangażowania grupy w realizację konkretnego etapu projektu i obecności oraz jakości wykonanych artefaktów. Maksymalna liczba punktów do zdobycia zależy od tematyki(symbolu) zajęć:

- a. P max. 10 p.
- b. I max. 5 p.
- c. R max. 25 p.
- d. O max. 15 p.

5. Harmonogram zajęć:

Nr	Symbol	Opis zajęć / zadania do wykonania
1	-	Zajęcia organizacyjne. Szkolenie BHP.
2	P	<ul style="list-style-type: none"><li>- przedstawienie nazwy i tematyki projektu, opis problemów biznesowych wymagających rozwiązania,</li><li>- przedstawienie podstawowych wymagań funkcjonalnych aplikacji,</li><li>- prezentacja stosu technologicznego do realizacji aplikacji <b>wraz z uzasadnieniem</b>, składającego się z co najmniej:<ul style="list-style-type: none"><li>- technologii utrwalania danych (RDBMS, NoSQL, ...),</li><li>- języka programowania aplikacji /frameworka po stronie "backendowej" (np. Django, node.js, RoR),</li><li>- technologii programowania aplikacji po stronie "frontendowej" (np. Vue, Angular, React),</li></ul></li></ul>

		<ul style="list-style-type: none"> <li>- systemu zarządzania wersjami (np. git),</li> <li>- serwera HTTP (np. Apache, nginx).</li> </ul>
3	P	<ul style="list-style-type: none"> <li>- przedstawienie architektury aplikacji (komunikacja frontend-backend, REST, mikroserwisy, MVC, ...),</li> <li>- zamodelowanie składowej bazodanowej (E-R) oraz przedstawienie modelu danych (UML - cd)</li> <li>- zamodelowanie min. 3 nietrywialnych procesów biznesowych (BPMN) implementowanych w aplikacji,</li> <li>- zaprojektowanie makiet/mockupów interfejsu aplikacji .</li> </ul>
4	I	<ul style="list-style-type: none"> <li>- przedstawienie raportu z postępów prac implementacyjnych</li> </ul>
5	I	
6	I	
7	I	
8	I	
9	I	
10	I	<ul style="list-style-type: none"> <li>- przedstawienie działania i kodu zaimplementowanej aplikacji</li> </ul>
11	R	<ul style="list-style-type: none"> <li>- przedstawienie planu eksperymentu badającego wydajności aplikacji uwzględniającego co najmniej 3 testy <i>end-to-end</i> dla zaimplementowanych funkcjonalności,</li> <li>- wykonanie skryptu automatyzującego opracowane scenariusze użycia aplikacji : <ul style="list-style-type: none"> <li>- przeprowadzenie eksperymentu,</li> <li>- wygenerowanie raportów.</li> </ul> </li> </ul>
12	R	<ul style="list-style-type: none"> <li>- przedstawienie raportów z badań oraz ich analizy zawierającej próbę wskazania wąskich gardeł aplikacji.</li> </ul>
13	O	<ul style="list-style-type: none"> <li>- implementacja mechanizmów optymalizujących działanie</li> <li>- przykładowe (choć nie jedyne) pomysły na optymalizację działania aplikacji: <ul style="list-style-type: none"> <li>- optymalizacja bazy danych (np. założenie indeksów),</li> <li>- analiza złożoności algorytmów realizujących procesy biznesowe i ich optymalizacja,</li> <li>- implementacja systemu ładowania zasobów "na żądanie" (ang. <i>lazy loading</i>),</li> <li>- zastosowanie systemu cache'owania (np. redis),</li> <li>- minifikacja i uglifikacja kodów serwowanych użytkownikowi,</li> <li>- optymalizacja konfiguracji serwera HTTP.</li> </ul> </li> </ul>
14	O	<ul style="list-style-type: none"> <li>- ponowne przeprowadzenie eksperymentu na zmodyfikowanej i analiza poprawy wydajności aplikacji</li> </ul>
15	-	Zajęcia końcowe. Publiczna prezentacja opracowanych aplikacji oraz wystawienie ocen.

6. Punkty zdobywają tylko ci członkowie grupy, którzy są obecni na zajęciach. Dodatkowo, prowadzący zastrzega sobie możliwość przyznania odmiennej liczby punktów poszczególnym

członkom grupy, jeżeli z rozmowy wyniknie odmienna znajomość postępów projektu.

Nie ma możliwości zdobywania punktów za zaległe zajęcia.

7. Dopuszczone są dwie nieusprawiedliwione nieobecności na zajęciach. Każda kolejna skutkuje obniżeniem oceny końcowej o 0.5.
8. Z przedstawionego harmonogramu wynika, że maksymalna, możliwa do zdobycia liczba punktów w ramach kursu wynosi:

$$2P + 7I + 2R + 2O = 20 + 35 + 50 + 30 = 135 \text{ pkt}$$

Otrzymana ocena jest proporcjonalna do liczby otrzymanych punktów wg procentowego ujęcia w tabeli (wartości punktów zaokrąglono do najbliższej całkowitej liczby):

%	pkt	Ocena
< 50	<68	2
50-60	68-81	3
60-70	81-95	3.5
70-80	95-108	4
80-90	108-122	4.5
90-98	122-132	5
98-100	132-135	5.5