

Projektowanie oprogramowania – Podgrupa1

I. Opis biznesowy „świata rzeczywistego” w języku klienta – aplikacja „Zapisy na zajęcia”

1. Opis zasobów ludzkich

1.1. Pracownik Uczelni, zarządzający zasobami systemu zapisów: wprowadzane dane dotyczące zasobów Uczelni:

Wydział	Stopień studów	Typ studiów:	Kierunek	Specjalność	Prowadzący zajęcia	Kurs-główna forma	Kurs-forma cząstkowa	Grupa - wykład	Grupa-laboratorium	Grupa-ćwiczenie	Grupa-seminarium	Grupa - projekt	Student	Wykaz zajęć studenta	Dane sali	Termin zajęć	Termin zapisów
Lista wydziałów	I i II	stacjonarne, niestacjonarne	Kierunki na wydziałach	Specjalności na kierunkach	Lista nauczycieli akademickich	Główna forma – do protokołu	Kurs cząstkowy	Dane grupy	Dane grupy	Dane grupy	Dane grupy	Dane grupy	Dane studenta		Budynek, nr sali	Terminy zajęć	

1.2. Prowadzący zajęcia, studenci

Prowadzący może prowadzić zajęcia tylko w jednej grupie w danym terminie. Student nie powinien zapisać się do dwóch różnych grup, których czas zajęć się pokrywa. Liczność grup powinna być ograniczona. System powinien wysyłać powiadomienia do studenta o terminach zapisów. Po przekroczeniu terminu student nie może zapisać się na zajęcia. Odblokowanie wymaga zgody ze strony Uczelni. Grupy nieobsadzone są likwidowane. Podobnie, jeśli w grupie jest zbyt mało studentów, grupa może być zlikwidowana wtedy, gdy studenci mogą być przydzieleni do innych grup - pod warunkiem, że nie zostanie przekroczona górna granica licznosci grupy. Podczas zapisów należy sprawdzić, czy student może zapisać się na zajęcia np kontrolując jego przynależność do grupy studentów z takimi uprawnieniami.

2. Przepisy

Podczas wyboru grupy student musi kierować się regulaminem studiów.

3. Dane techniczne

Należy zrealizować system z wykorzystaniem technologii Java EE, gdyż Uczelnia posiada dział wspierający utrzymanie oprogramowania w tej technologii. Dane dotyczące wydziałów, przedmiotów oraz kursów, są stabilne tzn nie wymagają częstych zmian. Zapisy odbywają się głównie w okresie poprzedzającym kolejny semestr. Uczelnia ma kilka pomieszczeń, w których mogą odbywać się zapisy.

II. Lista wymagań funkcjonalnych (wraz z minimalnym zestawem atrybutów)

1. Dodawanie konta studenta jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
2. Dodawanie wydziału jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
3. Dodawanie stopnia studiów jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
4. Dodawanie kierunku studiów (należy podać atrybuty, należy podać sposób identyfikacji)
5. Dodawanie specjalności jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
6. Dodawanie prowadzącego zajęcia jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
7. Dodawanie kursu głównego: wykład, seminarium, projekt jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
8. Dodawanie form towarzyszących kursowi głównemu: laboratorium, ćwiczenia, projekt, seminarium jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
9. Dodanie grup jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
10. Dodawanie terminów zajęć jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
11. Dodawanie terminów zapisów jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
12. Dodawanie danych pomieszczeń jako zasobu Uczelni (należy podać atrybuty, należy podać sposób identyfikacji)
13. Dodanie przydziału kursów do grup zajęciowych, pomieszczeń i terminów zajęć
14. Dodawanie wykazu zajęć studenta wynikający z typu i rodzaju studiów, roku studiów, wydziału, kierunku, specjalności oraz kursów
15. Dodawanie zapisu studenta na obowiązkowe i wybieralne zajęcia oparty na wyborze: konta studenta, jego wykazu zajęć, grup zajęciowych wynikających z wykazu zajęć
16. *Analiza zapisów (dane wejściowe do ustalenia, zastosowanie wybranego algorytmu typu Data mining)

III. Lista wymagań нефункциональных (do opracowania)

1. Liczba poszczególnych danych – studentów, zapisów,
2. Liczba grup i sal zajęciowych
3. Ograniczenia wydajnościowe
4. Czy jest wymagany masowy dostęp (Internet)?
5. Proponowane technologie

IV. Lista przypadków użycia - propozycja . Harmonogram prac poszczególnych sprintów zostanie podany w osobnym pliku

Tabela 1. Sprint 1 – dodawanie zasobów biura Uczelni (4 tygodnie)

Podgrupy 1 osoba jako Scrum Master do pomocy w poszczególnych podgrupach	Przypadki użycia – model, implementacja (logika biznesowa i GUI SE), testy: jednostkowe, akceptacyjne
1-a podgrupa (2 osoby)	1. PU Dodawanie wydziału 2. PU Dodawanie typu i rodzaju studiów, powiązanie z wydziałem 3. PU Dodawanie konta studenta i powiązanie go z wydziałem
2-a podgrupa (3 osoby)	1. PU Dodawanie kierunków na poszczególnych latach studiów 2. PU Dodawanie specjalności i powiązanie z kierunkami studiów 3. PU Dodawanie kursów oraz form cząstkowych 4. PU Dodawanie terminów zajęć
3-podgrupa (2 osoby)	1. PU Dodawanie grup zajęciowych 2. PU Dodawanie sal zajęciowych 3. Dodawanie prowadzących zajęcia

Tabela 2. Harmonogram realizacji 1 sprintu (tabela 1)

Opis realizacji sprintu dla trzech podgrup zespołu					
Nr tygodnia Semestru/ nr tygodnia sprintu	Sprint	Spotkanie	Uwagi dotyczące realizacji zadań przez każdą z dwóch podgrup zespołu	Liczba punktów (do oceny)	Zadania Scrum Master
2 /1	1	Sprint planning meeting (90 min)	Zajęcia organizacyjne (podział na grupy i podgrupy, przydzielenie ról projektowych, uzyskanie dostępu do wymaganych narzędzi) <ul style="list-style-type: none"> • User Stories – Analiza dostarczonego modelu biznesowego „świata rzeczywistego” systemu– <i>udział wszystkich grup projektowych</i>. Każda grupa projektowa otrzymuje ogólny opis procesów biznesowych, ale może opracować dokładnie wybrany fragment opisu biznesowego • Sprint Backlog (formy pośrednie: Product Backlog), Sprint planining) - ewnetualna dostarczonych modyfikacja wymagań funkcjonalnych i niefunkcjonalnych– <i>udział wszystkich grup projektowych</i>. Każda grupa dokładnie weryfikuje część wymagań funkcjonalnych wynikających z otrzymanego fragmentu opisu „świata rzeczywistego” 	3-5	Współdziałanie z wykonawcami z podgrup

			<ul style="list-style-type: none"> • Definicja PU (przypadku użycia): opis słowny wg standardowego formularza – scenariusz należy wykonać za pomocą diagramu aktywności.. Każda grupa opracowuje przypadki użycia jako specyfikację tych wymagań funkcjonalnych, które opracowała w poprzednich krokach. Można wykonać kod wg scenariuszy PU, jeśli został zidentyfikowany scenariusz działania, analogiczny jak w przypadku dodawania obiektów typu TTitle (instrukcje do lab1) oraz obiektów TRachunek lub TProdukt1 i TProdukt2 w instrukcji do lab8. <p>Wyniki prac są umieszczane:</p> <ul style="list-style-type: none"> • projektu Registration for classes w repozytorium Repository_team1, • projektu Travel agency w repozytorium Repository_team2, <p>i zostaną ocenione oceniane przez prowadzącego zajęcia.</p>		
3/2	1	Daily Scrum of Scrums (20 min)	<p>Przedstawienie wyników prac grup z 1 tygodnia (wersja początkowa). Projekty zawierają:</p> <ol style="list-style-type: none"> 1) diagram przypadków użycia, 2) diagramy aktywności 3) kod na podstawie scenariuszy PU i rozwiązań podanych w instrukcjach do lab1 i lab8. P. Inżynieria Oprogramowania <p>Podczas 2-tygodnia wyniki prac umieszczane są w repozytoriach</p> <ol style="list-style-type: none"> 1) Podgrupa 1 – wykonanie kodu (cd) i GUI dla jednego PU, testy jednostkowe i akceptacyjne implementowanego PU 2) Podgrupa2 (2osoby) – wykonanie kodu (cd) i GUI dla jednego PU, testy jednostkowe i akceptacyjne implementowanego PU 3) Podgrupa3 – wykonanie kodu i GUI dla jednego PU, testy jednostkowe i akceptacyjne implementowanego PU <p>Przykład połączenia wartswy klienta (GUI – projekt Library1_client1_SE) i logiki biznesowej (projekt Library1): Przykład programu</p>	3-5	<p>Scrum Master i jeden student z podgrupy 2:</p> <ul style="list-style-type: none"> • integrują diagramy przypadków użycia, korygują scenariusze przypadków użycia, • tworzą diagram wymagań • dodają diagramy aktywności <p>W wyniku ma powstać jeden projekt UML integrujący diagramy przypadków użycia trzech grup jako jeden diagram, diagramy aktywności wykonane podczas 2 tygodnia, diagram wymagań</p>
4/3	1	Daily Scrum of Scrums (20 min)	<p>Przedstawienie wyników prac grup z 2 tygodnia oraz Scrum master i wybranej osoby z podgrupy 2. Prezentacja obejmuje:</p> <ol style="list-style-type: none"> 1) DPU prezentujący zintegrowany DPU oraz wykonane diagramy aktywności i diagram wymagań 2) Kody trzech podgrup zawierające dwie warstwy: klienta (GUI – podobnie jak projekt Library1_client1_SE) i logiki biznesowej (podobnie jak projekt Library1), prezentacja wyników testów 	3-5	<p>Współdziałanie z wykonawcami z podgrup</p> <p>Wykonanie diagramu klas i dodanie do projektu UML na podstawie klas wykorzystanych w kodzie trzech podgrup</p>

			<p>Podczas 3-tygodnia wyniki prac umieszczane są w repozytoriach</p> <ol style="list-style-type: none"> 1) Podgrupa 1 – wykonanie kodu (cd) i GUI – kontynuacja (2-i PU), testy jednostkowe i akceptacyjne implementowanego PU 2) Podgrupa2 (3 osoby) – wykonanie kodu i GUI – kontynuacja (2-i i 3-i PU), testy jednostkowe i akceptacyjne implementowanych PU 3) Podgrupa3 – wykonanie kodu i GUI—kontynuacja (3-i PU), testy jednostkowe i akceptacyjne implementowanego PU 		
5/4	1	Daily Scrum of Scrums (20 min)	<p>Przedstawienie wyników prac grup z 2 tygodnia oraz Scrum master i wybranej osoby z podgrupy 2. Prezentacja obejmuje:</p> <ol style="list-style-type: none"> 1) DPU prezentujący diagram klas 2) Kody trzech podgrup zawierające dwie warstwy: klienta (GUI – podobnie jak projekt Library1_client1_SE) i logiki biznesowej (podobnie jak projekt Library1) <p>Podczas 4-tygodnia wyniki prac umieszczane są w repozytoriach</p> <ol style="list-style-type: none"> 1) Podgrupa 1 – wykonanie kodu (cd) i GUI – kontynuacja (3-i PU), testy jednostkowe i akceptacyjne implementowanego PU 2) Podgrupa2 (3 osoby) – wykonanie kodu i GUI – kontynuacja (4-y PU), testy jednostkowe i akceptacyjne implementowanych PU 3) Podgrupa3 – wykonanie kodu i GUI—kontynuacja (3-i PU), testy jednostkowe i akceptacyjne implementowanego PU 	3-5	Współdziałanie z wykonawcami z podgrup

Sprint 2 – dodawanie powiązań pomiędzy danymi (4 tygodnie)

Podgrupy 1 osoba jako Scrum Master do pomocy w poszczególnych podgrupach	Przypadki użycia – model, implementacja (logika biznesowa, GUI EE, JPA), testy: jednostkowe, akceptacyjne, funkcjonalne
1-a podgrupa – 4 osoby	1. PU Powiązanie kursów z kierunkami i/lub specjalnościami 2. PU Powiązanie prowadzących zajęcia z grupami, kursami i terminami
2-a podgrupa – 3 osoby	1. PU Dodawanie wykazu zajęć studenta jako wykaz kursów

Sprint 3 – dodawanie powiązań pomiędzy danymi (cd – 3 tygodnie)

Podgrupa 1 osoba jako Scrum Master do pomocy w poszczególnych podgrupach	Przypadki użycia – model, implementacja (logika biznesowa, GUI EE JPA), testy: jednostkowe, akceptacyjne, funkcjonalne
1-a podgrupa	1. PU Integracja wyników prac grupy 1 i 2 ze sprintu 2 2. PU Dodawanie zapisów na zajęcia
2-a podgrupa	2. *PU Analiza danych dotyczących zapisów

Sprint 4 – kontynuacja implementacji (3 tygodnie)

Podgrupa 1 osoba jako Scrum Master do pomocy w poszczególnych podgrupach	Przypadki użycia - kontynuacja implementacji na platformie Java EE oraz testów
1-a podgrupa	1. PU Dodawanie zapisów na zajęcia (cd)
2-a podgrupa	2. *PU Analiza danych dotyczących zapisów (cd)