



UNIwersytet
Przyrodniczy
we Wrocławiu

Diagram przypadków użycia

Wprowadzenie do Inżynierii Oprogramowania
2018/2019

Radosław Zajdel
Instytut Geodezji i Geoinformatyki, pok. 329a
radoslaw.zajdel@upwr.edu.pl

Klasyfikacja wymagań BABOK 3.0

BABOK (ang. A Guide to the Business Analysis Body of Knowledge) to zbiór dobrych praktyk z zakresu analizy biznesowej. Jest to dzieło IIBA (ang. International Institute of Business Analyst) – międzynarodowej organizacji zrzeszającej profesjonalistów i entuzjastów analizy biznesowej.

- **Wymagania Biznesu** – Potrzeby klientów
 - **Wymagania Interesariuszy** – Co poszczególne osoby, jakkolwiek „zaangażowane” w projekt wymagają od oprogramowania
 - **Wymagania na Rozwiązanie/System** - Wymagania te realizują wymagania użytkowników. Stanowią rozwiązanie problemu lub realizują cel →
-

Klasyfikacja wymagań

- **wymagania funkcjonalne** – dotyczące tego co ma realizować system; jakie ma spełniać funkcje, jakich dostarczać usług, jak zachowywać się w określonych sytuacjach. Powinny być kompletne (wszystkie usługi) i spójne (sprzeczności).
 - Funkcje systemu widziane od strony użytkownika: zakup książki, drukowanie raportu, przybliżenie mapy
 - **wymagania niefunkcjonalne** – dotyczące tego jak system powinien realizować swoje zadania; np. wymagania dotyczące koniecznych zasobów, ograniczeń czasowych, niezawodności, bezpieczeństwa, przenośności, współpracy z określonymi narzędziami i środowiskami, zgodności z normami i standardami, a także przepisami prawnymi, w tym dotyczącymi zabezpieczeń i prywatności, itp.
-

<Project Name>	
Wizja	Data: <dd.mm.yyyy>

<Project Name> Wizja

Usage note: There is procedural guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute allowing you to toggle whether it is visible or hidden in this template. Use the Word menu Tools→Options→View→Hidden Text checkbox to toggle this setting. A similar option exists for printing Tools→Options→Print.

1. Wprowadzenie

[General description of the product]

2. Pozycjonowanie

2.1 Sformułowanie problemu

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

Problem	[describe the problem]
Dotyczy	[the stakeholders affected by the problem]
Wpływ problemu	[what is the impact of the problem?]
Pomysłowe rozwiązanie	[list some key benefits of a successful solution]

2.2 Opis pozycji produktu

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

Dla	[target customer]
Który	[statement of the need or opportunity]
(Nazwa produktu)	is a [product category]
Który	[statement of key benefit; that is, the compelling reason to buy]
Inaczej niż	[primary competitive alternative]
Nasz produkt	[statement of primary differentiation]

[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

3. Opis udziałowców i użytkowników

3.1 Podsumowanie udziałowców

Nazwa	Opis	Odpowiedzialności
[Name the stakeholder type.]	[Briefly describe the stakeholder.]	[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder: ensures that the system will be maintainable ensures that there will be a market demand for the product's features]

<Project Name>	
Wizja	Data: <dd.mm.yyyy>

Nazwa	Opis	Odpowiedzialności
		monitors the project's progress approves funding and so forth]

3.2 Podsumowanie użytkowników

Nazwa	Opis	Odpowiedzialności
[Name the stakeholder type.]	[Briefly describe the stakeholder.]	[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder: ensures that the system will be maintainable ensures that there will be a market demand for the product's features monitors the project's progress approves funding and so forth]

4. Opis produktu

4.1 Potrzeby i cechy

[Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented. Capture the stakeholder priority and planned release for each feature.]

Potrzeba	Priorytet	Cechy	Planowane wydanie

4.2 Inne wymagania produktowe

[At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements.

Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.

Note any design constraints, external constraints, assumptions or other dependencies that, if changed, will alter the Vision document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the Vision document will need to change.

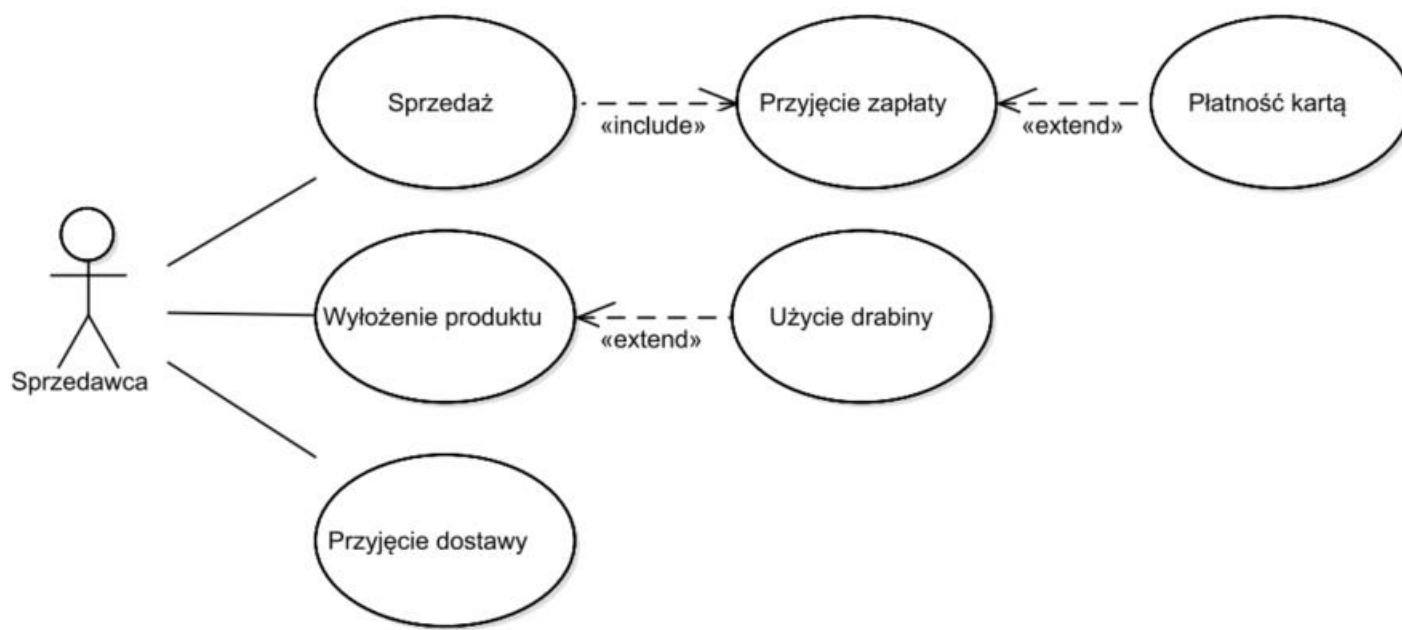
Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.

Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.]

Wymaganie	Priorytet	Planowane wydanie

Diagram Przypadków Użycia

- Diagram przypadków użycia to graficzne przedstawienie przypadków użycia funkcji systemu, aktorów oraz związków między nimi, występujących w danej dziedzinie przedmiotowej.



Cel diagramów przypadków użycia

- Główne cele
 - Analiza obszaru zastosowań, dziedziny przedmiotowej
 - Definicja **roli aktora wobec systemu**
 - Określenie **usług świadczonych przez system na rzecz aktorów** (użytkowników, klientów)
 - Stanowią podstawę testowania funkcji systemu na dalszych etapach jego cyklu życia
 - Nie pokazuje **pełnej informacji o systemie wiele rzeczy może zostać spakowanych w jeden use case !**
 - **Opis ważniejszy niż diagramy**
-

Dygresja – rodzaje przypadków użycia

- Przypadki użycia, które odnoszą się do procesów biznesowych, nazywamy biznesowymi przypadkami użycia
 - Przypadki użycia, które odnoszą się do oprogramowania, nazywamy systemowymi przypadkami użycia
-

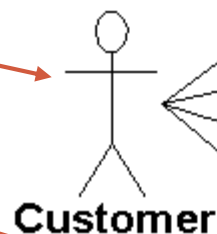
Podstawowe pojęcia

Diagram przypadków użycia

- Przypadek użycia

- Aktor

- Związki



Open Account

Deposit Funds

Withdraw
Funds

Close Account

Przypadek użycia

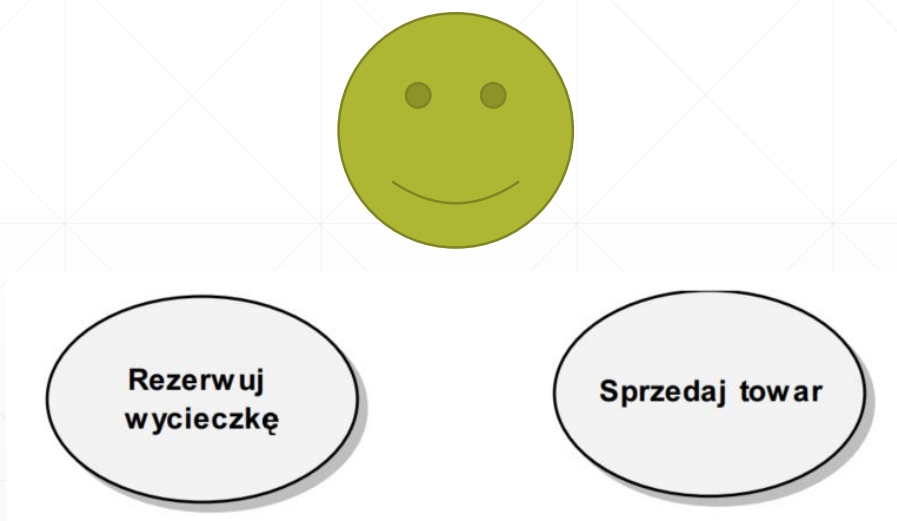
- Przypadek użycia (use case) – specyfikacja ciągu akcji i ich wariantów, które system (lub inna jednostka) może wykonać poprzez interakcje z aktorami tego systemu.

Działanie realizowane w systemie w konsekwencji aktywności aktora

- Najpowszechniejszy sposób definiowania wymagań dla system informatycznego. Możliwe jest jego utworzenie na podstawie opisu systemu (dokument wizji)
 1. Opis interakcji aktora z systemem prowadzący do konkretnego celu
 2. Brak szczegółów związanych z technologią i interfejsem użytkownika
 3. Odpowiednio dobrany poziom szczegółowości (nie rozdrabniamy się jeżeli nie ma takiej potrzeby)
-

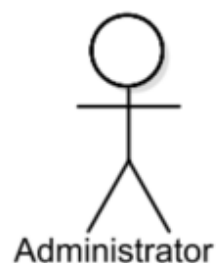
Przypadek użycia

- Nazwa stanowi zwięzłe polecenie wykonania funkcji, najczęściej sformułowanie w trybie rozkazującym.



Aktor

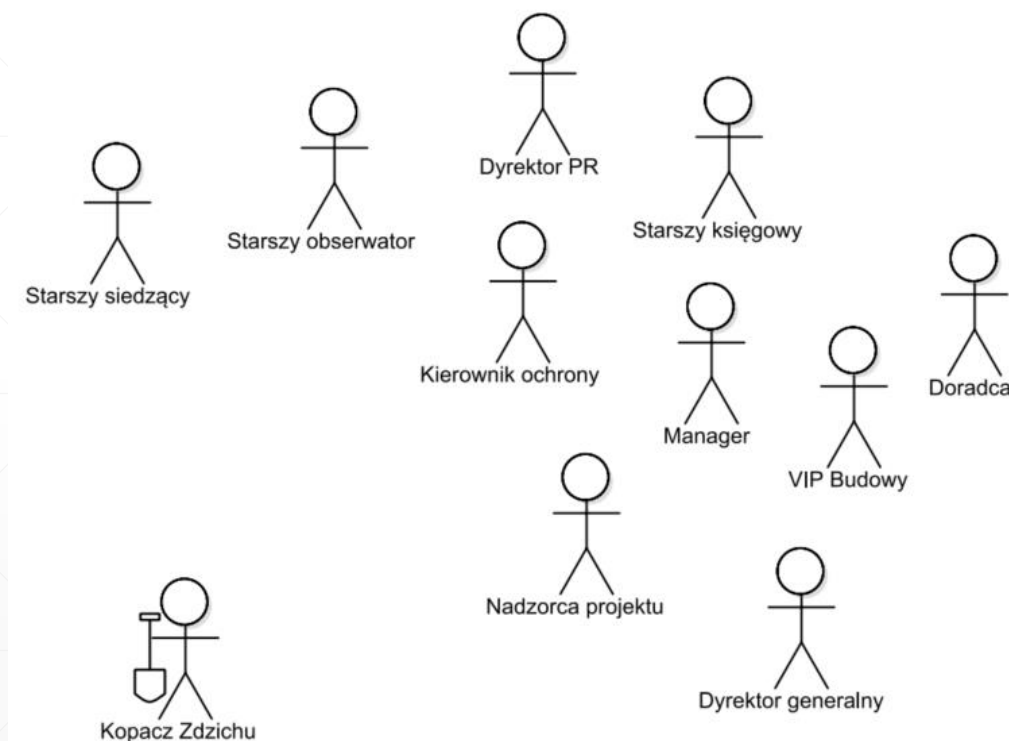
- Aktor jest to spójny zbiór ról odgrywanych przez użytkowników przypadków użycia w czasie interakcji z tym przypadkiem użycia. Odnosi się do diagramu klas ale nie musi się z nim pokrywać.
- Najczęściej aktorami są użytkownicy i klienci systemu, ale możemy wyróżnić aktorów **osobowych i nieosobowych**



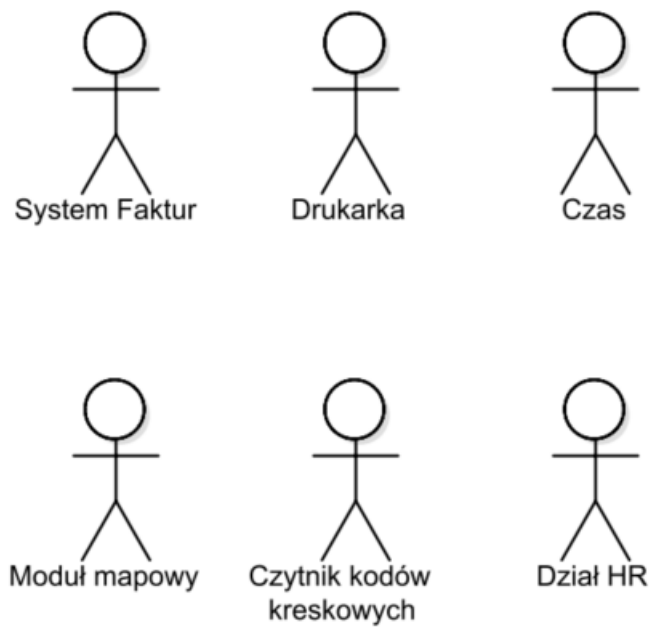
Rozpoznanie aktorów systemu

- Rolę aktora osobowego może pełnić osoba, zespół, dział, instytucja, organizacja, zrzeszenie organizacji.
 - Nazwy aktorów powinny być jednoznaczne
 - Nazwy aktorów osobowych najczęściej pokrywają się z nazwami stanowisk lub pełnionej funkcji w „systemie”/”organizacji”/”projekcie”
 - Nazwy aktorów nieosobowych to najczęściej systemy zewnętrzne (podsystemy, bazy danych), urządzenia oraz czas
-

Rozpoznanie aktorów systemu

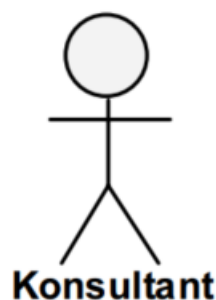


Aktorzy nieosobowi



Aktorzy cd.

- Nazwa aktora – rzeczownik lub wyrażenie rzeczownikowe w l. pojedynczej
- Identyfikując aktora pamiętamy że odzwierciedla ona rolę a nie indywidualną jednostkę/obiekt. Np.



Zakładamy że firma zatrudni wielu konsultantów, mimo to przypadek użycia dla każdego z nich będzie wyglądał schematycznie tak samo, a nazwa pozostanie w liczbie pojedynczej. Z punktu widzenia systemu każdy konsultant odgrywa tę samą rolę.

Interakcje Przypadek użycia - Aktor

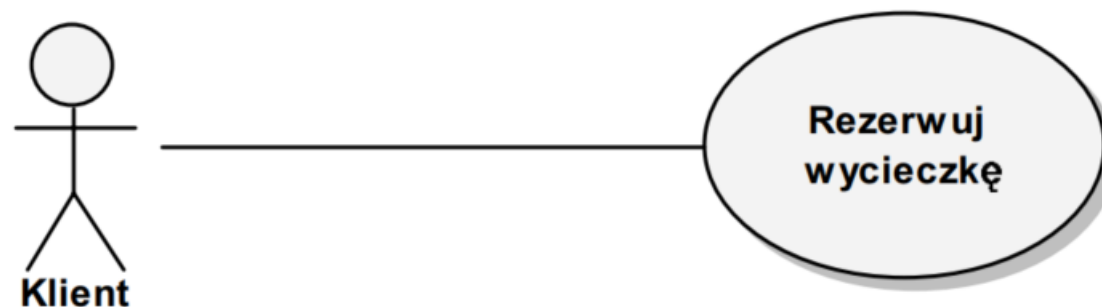
- Aktor może użytkować jeden lub więcej przypadków użycia w danym systemie
 - Przypadek użycia może być użytkowany przez jednego lub więcej aktorów
-

Związki

- Każdy aktor umieszczony na diagramie powinien być bezpośrednio powiązany z co najmniej jednym przypadkiem użycia
 - Analogicznie do poprzedniego wykładu możemy wyróżnić cztery rodzaje związków:
 - **Asocjację**
 - **Uogólnienie**
 - **Zależność**
 - **Realizację**
-

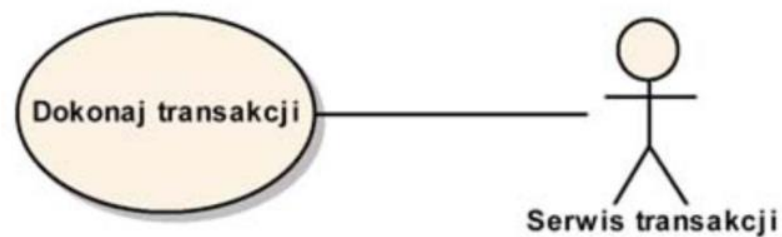
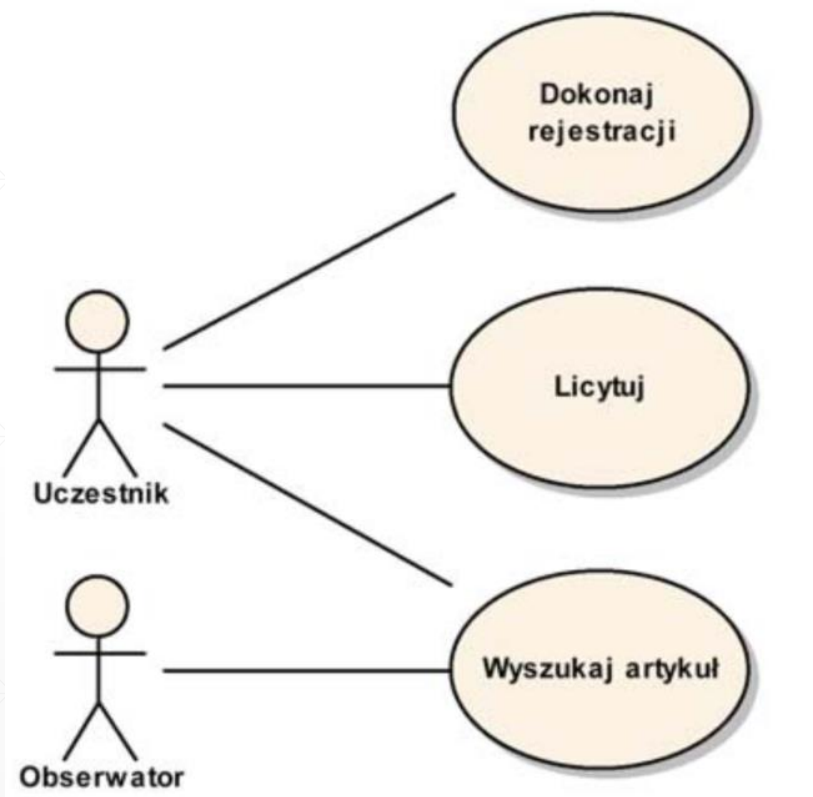
Asocjacja

- Dwukierunkowa komunikacja pomiędzy aktorem a przypadkiem użycia



- W odróżnieniu od diagramu klas, **nie formułuje się nazwy związku**
-

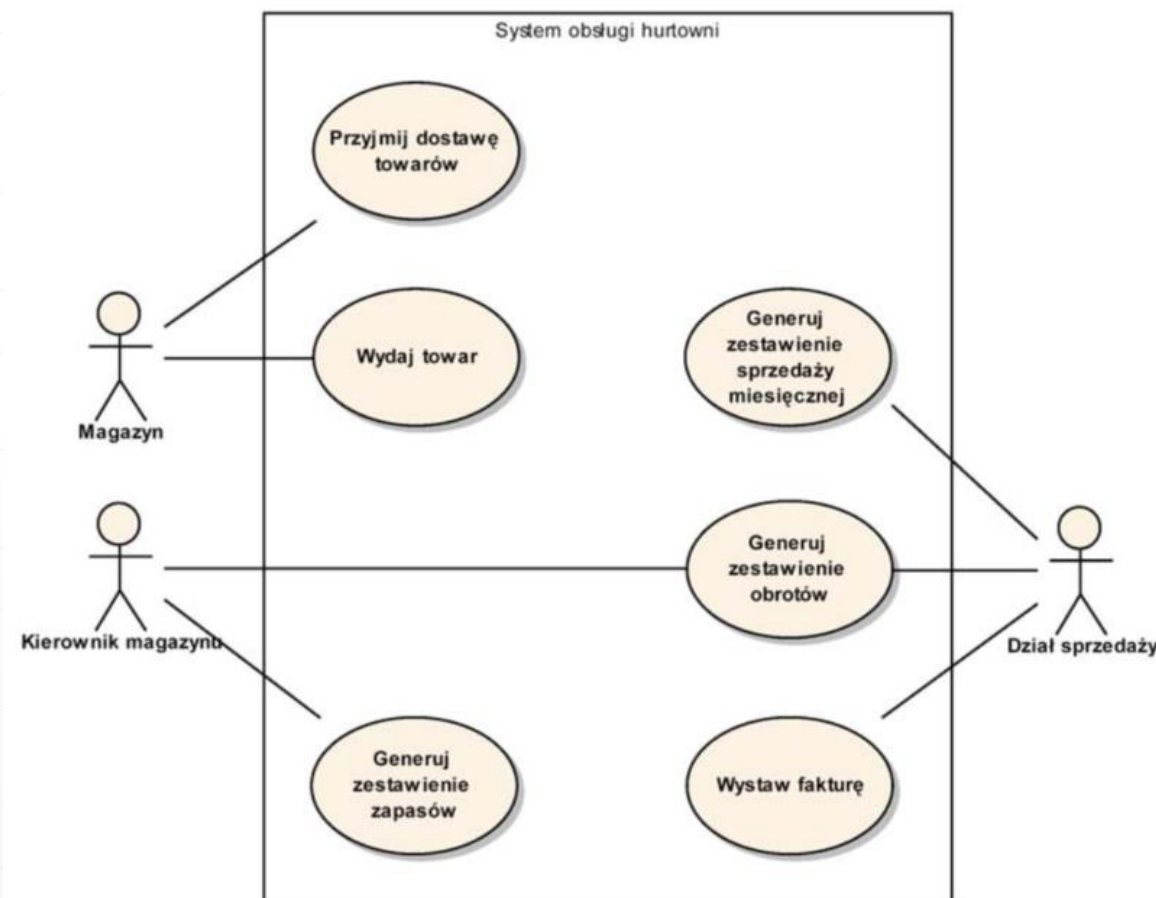
Przykład



- 3 aktorów (jeden zewnętrzny)
- Tylko zarejestrowany uczestnik może brać udział w aukcji ale każdy może wyszukać artykuł
- System komunikuje się również z serwisem transakcyjnym w celu poprawnego przeprowadzenia transakcji

Granica obszaru zastosowań

- Granica obszaru zastosowań reprezentuje zakres funkcjonalny przyszłego systemu.
- Aktorów umieszcza się po za granicą obszaru zastosowań

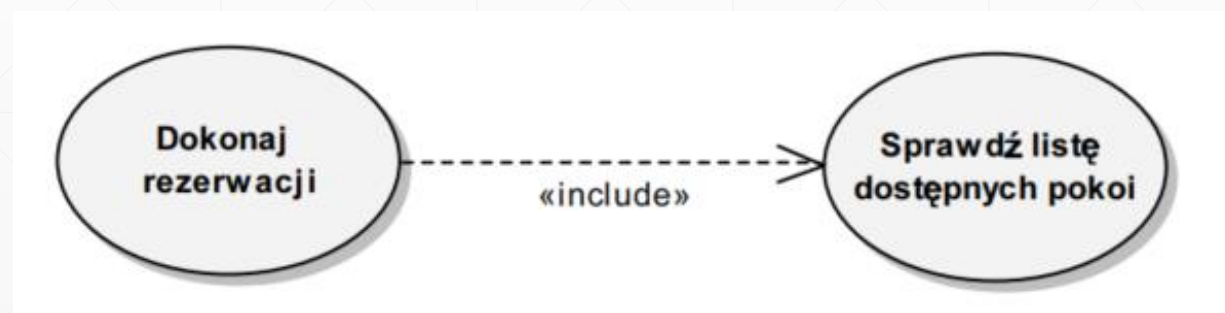


Zaawansowane składniki diagramu

- koncepcję diagramów PU można rozbudować do kolejnych uszczegółowień:
 - Wykorzystanie zależności **zawierania (include)**
 - Wykorzystanie zależności **rozszerzania (extend)**
 - **Uogólnienie**
 - **Rodzaj aktora**
 - **Liczebność, nawigację, realizację**
 - **Dokumentacje przypadków użycia**
-

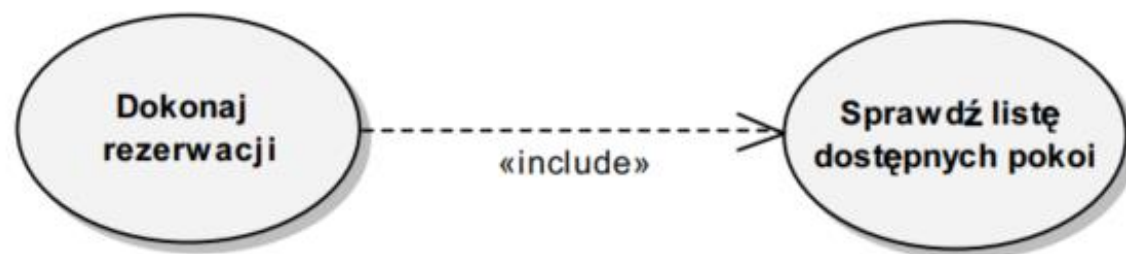
Zależność zawierania

- Oznaczany przez znacznik <<include>> umiejscowiony na asocjacji
- Powiązanie pomiędzy przypadkiem zawierającym (tj. bazowym), a przypadkiem zawieranym
- Związek **obligatoryjny** – tzn. **zawsze jest wykonywany**
- **Zawierany przypadek użycia nie jest wykonywany samodzielnie, lecz wyłącznie przy odwołaniu się do większego zawierającego przypadku użycia**
- **Skierowana od zawierającego do zawieranego**



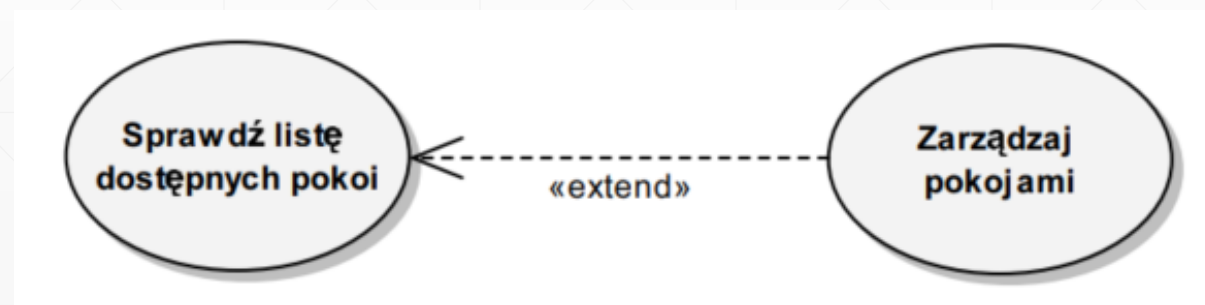
Kiedy tworzenie zależności *include* ma sens ?

1. Jeżeli istnieje możliwość wydzielenia wspólnej dla kilku przypadków użycia spójnej funkcjonalności (zasada *DRY*). Dzięki temu nie zachodzi konieczność powielania analogicznych treści
2. Interakcje pomiędzy aktorem a systemem w przypadku danego przypadku użycia są bardzo liczne



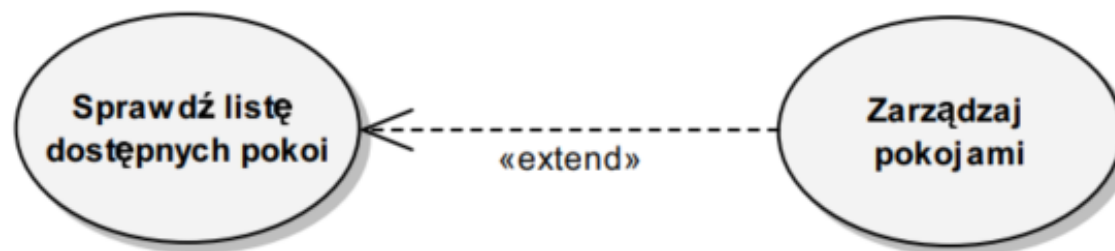
Zależność rozszerzania

- Oznaczany przez znacznik <<extend>> umiejscowiony na asocjacji
- Powiązanie pomiędzy przypadkiem rozszerzanym przypadkiem użycia (tj. bazowym), a przypadkiem rozszerzającym
- Związek **opcjonalny** – tzn. **zostaje wykonywany gdy zajdzie taka potrzeba**
- **Funkcjonalność rozszerzającego przypadku użycia może, ale nie musi zostać włączona do rozszerzanego przypadku użycia. Wymagane jest spełnienie jakiegoś warunku**
- **Skierowana od rozszerzającego do rozszerzanego**



Kiedy tworzenie zależności *extend* ma sens ?

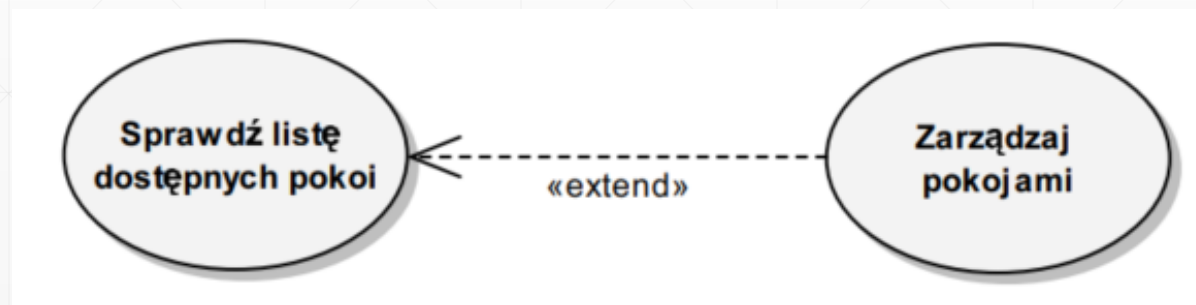
1. Funkcjonalność reprezentowana przez rozszerzany przypadek użycia ma zostać uzupełniona o kilka dodatkowych kroków. Nie są one jednak wykonywane automatycznie przy każdym odwołaniu, a jedynie w określonych sytuacjach



Extend

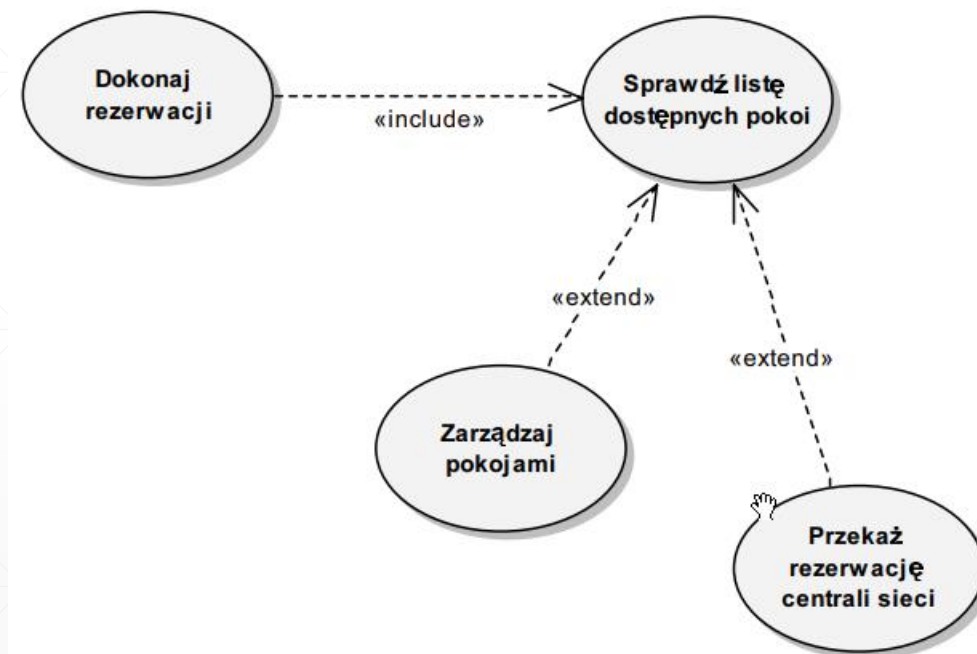
- Sprawdzenie listy pokoi nie jest jednoznaczne z Zarządzaniem pokojami. System hotelowy, oprócz wyszukiwania pokoi spełniających wyspecyfikowane przez klienta kryteria, musi bowiem uwzględniać takie elementy jak:
 - Przypisanie pokoju innej kategorii
 - Zmiana ceny wynajmu
 - Zmiana statusu pokoju
 - Czasowe wyłączenie pokoju z użytkownika

Zarządzanie pokojami

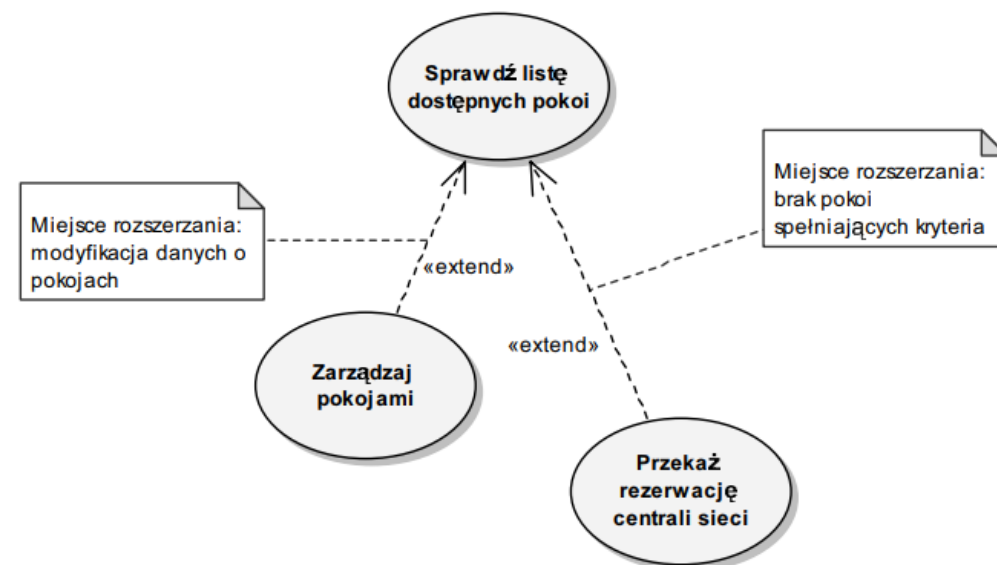
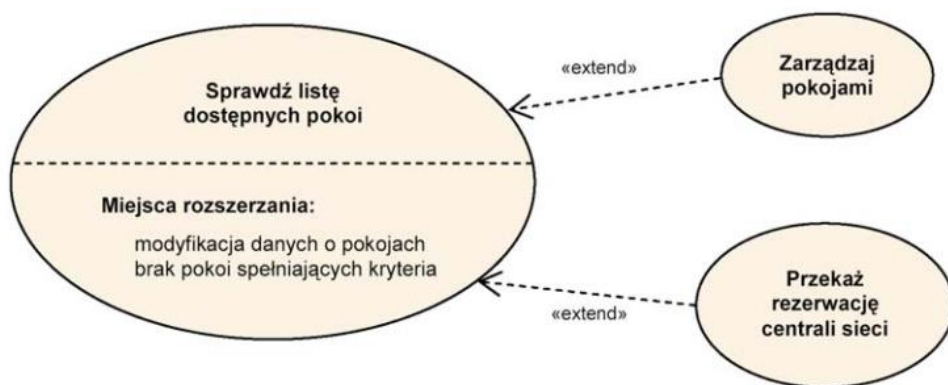


Zawieranie i rozszerzenie

- Zależności zawierania i rozszerzenia **występują niezależnie od siebie**
- Przypadek bazowy może być rozszerzany o przypadki rozszerzające po spełnieniu określonych warunków.
- Warunki te określa się w przypadku bazowym jako **miejsca rozszerzania**
- Po wykonaniu działań związanych z rozszerzeniem kontynuowane są działania należące do przypadku bazowego



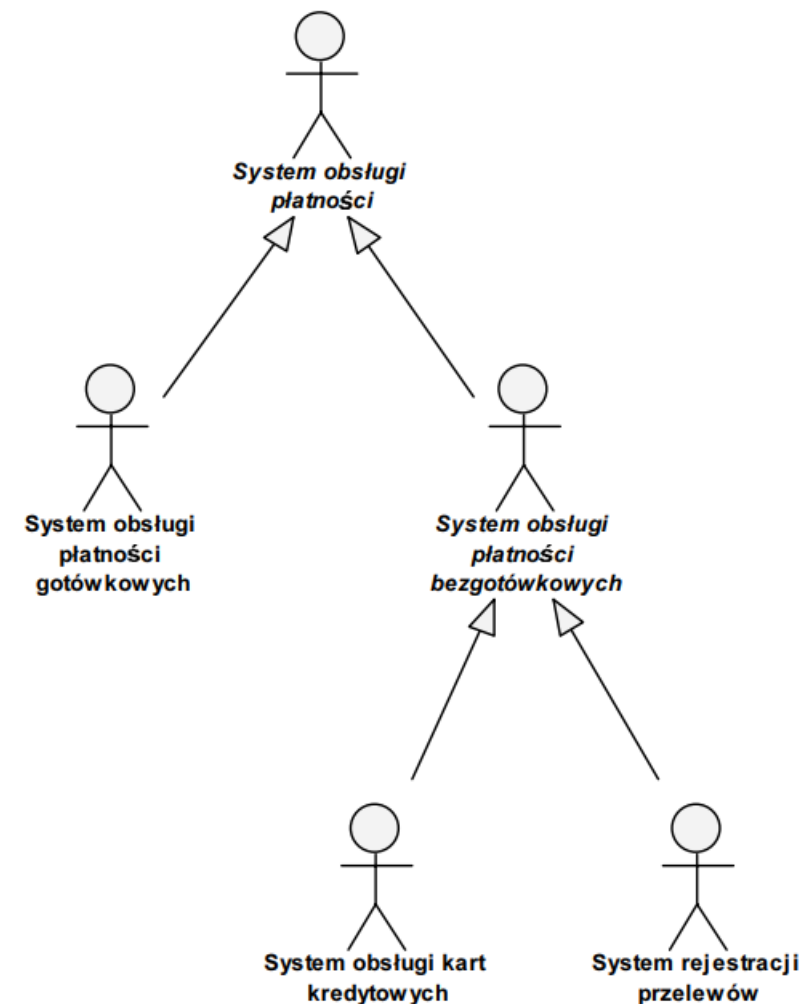
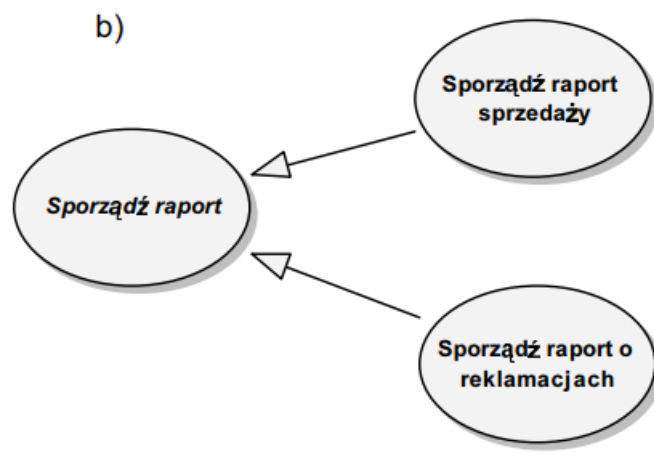
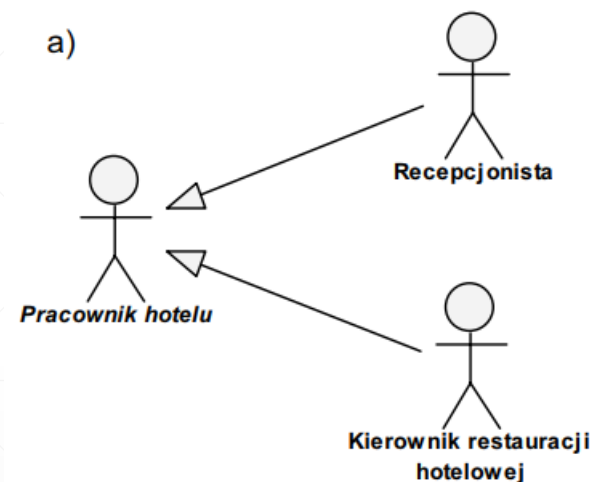
Przykład określenia miejsca rozszerzenia



Uogólnienia

- Uogólnienie to związek pomiędzy **klasyfikatorem ogólnym a specjalizowanym**
 - Element specjalizowany z założenia **dziedziczy wszelkie cechy elementu ogólnego.**
 - Związki uogólnienia **dotyczą zarówno przypadków użycia jak i aktorów**
-

Przykłady uogólnień



Rodzaje aktorów - stereotypy

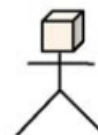
- Stereotypy graficzne aktorów

a)



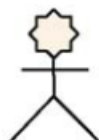
Klasyczna
reprezentacja –
człowiek

b)



System
zewnętrzny

c)



Urządzenie

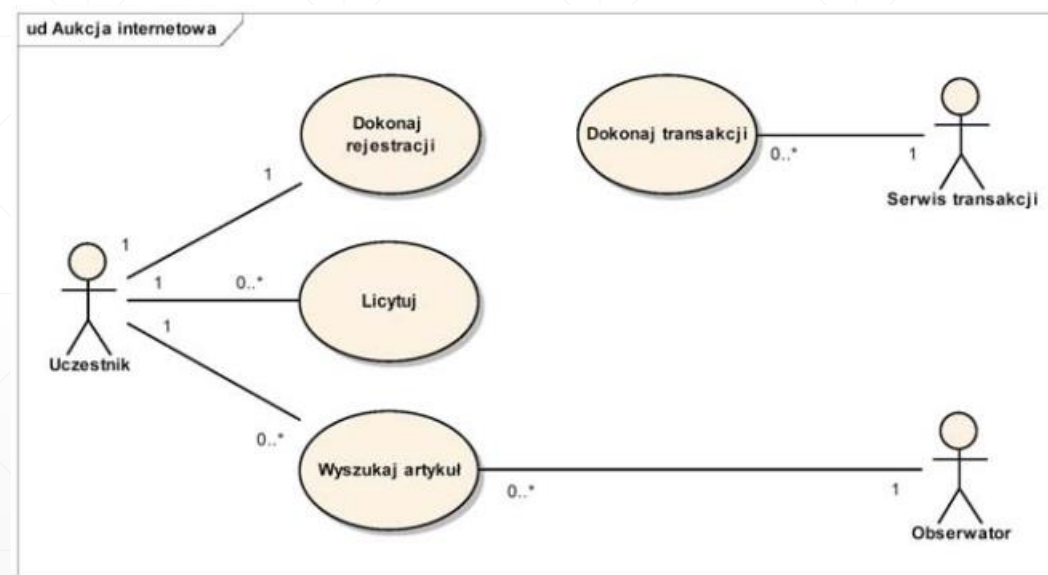
d)



Czas

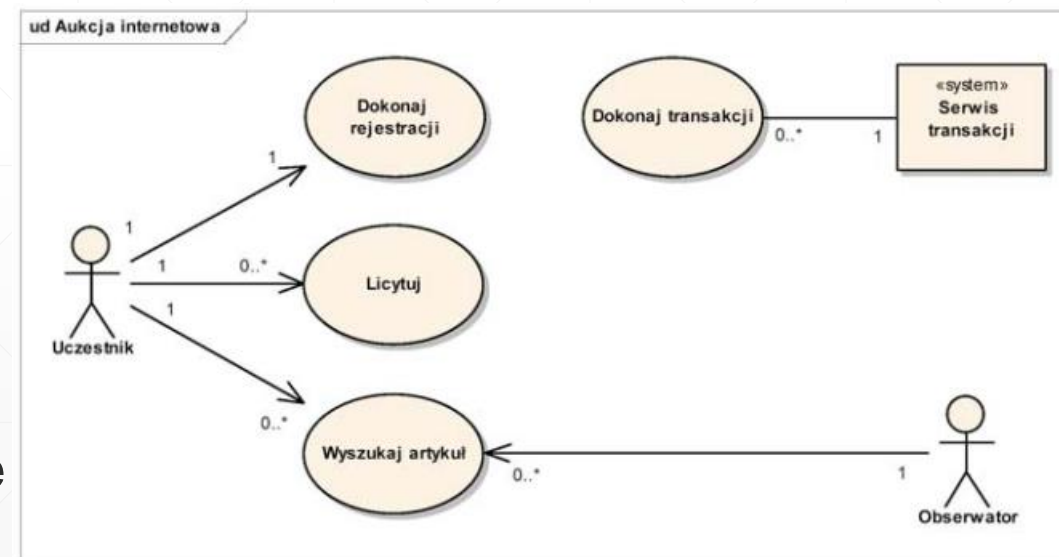
Liczebność

- Zasada ta sama co w przypadku **diagramu klas**
- Pojedynczy *Uczestnik*, związany jest z jednym i tylko jednym przypadkiem użycia *Dokonaj rejestracji*
- Funkcjonalności: *Wyszukiwanie artykułu*, *Licytowanie*, *Dokonywanie transakcji* w odniesieniu do pojedynczego aktora może być inicjowane wielokrotnie lub ani razu



Nawigacja

- Asocjacje są zawsze binarne (łączy dwie instancje/klasyfikatory)
- Związki reprezentują interakcje, a nie przepływ danych dlatego z zasady nie wykorzystuje się kierunków asocjacji
- W wyjątkowych przypadkach asocjacje można wzbogacić o wskazanie kierunku nawigacji w postaci strzałki
- Oznaczenie nawigacji jest stosowane tylko wtedy, gdy funkcja inicjatora przypadku użycia lub aktora jest istotna i wymaga udokumentowania na diagramie



Dokumentacja przypadków użycia

- Diagramy przypadków użycia przedstawiają bardzo ogólny obraz systemu, stanowiący punkt startowy jego tworzenia, lecz nie pozwalający na przedstawienie wielu istotnych informacji, które są niezbędne w realizacji dalszych faz cyklu życia systemu
 - Dla danego przypadku użycia należy wyróżnić **scenariusz główny** oraz zgodnie z potrzebą **scenariusze alternatywne**
 - **Scenariusz stanowi określony ciąg akcji dokumentujący zachowanie**
-

Dokumentacja przypadku użycia

- Sposoby dokumentacji przypadku użycia
 - Niesformalizowany tekst
 - Formalny tekst strukturalny
 - Pseudokod
 - Tabela
-

Opis dokumentacji tabelarycznej

Nazwa:	Pełna nazwa przypadku użycia
Numer:	Numer identyfikacyjny przypadku użycia
Twórca:	Dane twórcy przypadku użycia, np. imię, nazwisko, stanowisko
Poziom ważności:	Określenie poziomu ważności przypadku z perspektywy użytkownika, np. niski, średni, wysoki
Typ przypadku użycia:	Określenie typu przypadku użycia z punktu widzenia jego złożoności oraz ważności dla zaspokojenia potrzeb użytkownika, np. ogólny/szczegółowy; niezbędny/istotny/przeciętnie istotny/moło istotny
Aktorzy:	Lista aktorów będących w związku z przypadkiem użycia
Krótki opis:	Krótką, ogólną charakterystyką przypadku użycia
Warunki wstępne:	Charakterystyka koniecznych warunków inicjujących przypadek
Warunki końcowe:	Charakterystyka stanu systemu po realizacji przypadku użycia
Główny przepływ zdarzeń:	Wypunktowana i scharakteryzowana lista przepływów zdarzeń zachodzących podczas realizacji przypadku użycia; scenariusz główny
Alternatywne przepływy zdarzeń:	Wypunktowana i scharakteryzowana lista możliwych, alternatywnych przepływów zdarzeń przypadku użycia
Specjalne wymagania:	Wypunktowana i scharakteryzowana lista dodatkowych zidentyfikowanych wymagań niefunkcjonalnych, które mogą być istotne przykładowo podczas projektowania czy kodowania
Notatki i kwestie:	Lista wszelkich komentarzy dotyczących przypadku użycia i lista pozostałych otwartych kwestii, które powinny zostać rozwiązane wraz z propozycjami osób, które mogłyby je rozwiązać

Anuluj rezerwację

- Warunek wstępny
- Warunek końcowy
- Główny przepływ
- Alternatywny przepływ

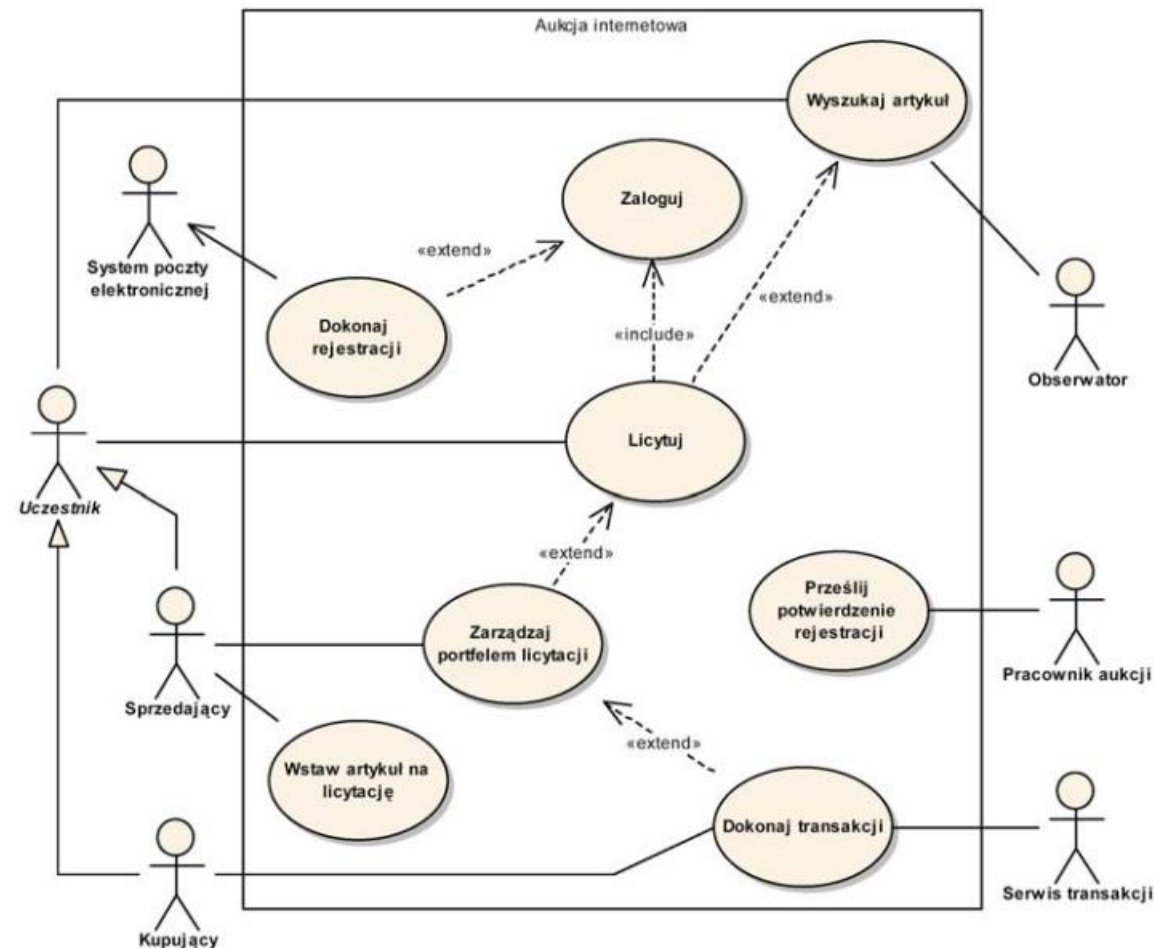
Nazwa:	Anuluj rezerwację
Numer:	5
Twórca:	Anna Krotoszyńska — Projektant
Poziom ważności:	Średni
Typ przypadku użycia:	Ogólny
Aktorzy:	Recepcjonista, Kierownik recepcji
Krótki opis:	Anulowanie istniejącej rezerwacji pokoju lub apartamentu
Warunki wstępne:	Co najmniej jeden pokój lub apartament hotelowy musi być zarezerwowany
Warunki końcowe:	System odnotowuje pokój lub (i) apartament jako dostępny
Główny przepływ zdarzeń:	<ol style="list-style-type: none">1. Recepcjonista weryfikuje rezerwację, uruchamiając funkcję „Rezerwacje”2. System wyświetla okno z informacjami o rezerwacjach (pokoje i apartamenty hotelowe)3. Pracownik recepcji zaznacza rezerwacje do anulowania i uruchamia funkcję „Anuluj rezerwacje”4. System wyświetla komunikat „Czy anulować zaznaczone rezerwacje?”5. Pracownik recepcji potwierdza operację anulowania zaznaczonych rezerwacji6. System potwierdza wykonanie operacji komunikatem „Anulowano wybrane rezerwacje” i odświeża ekran monitora
Alternatywne przepływy zdarzeń:	<ol style="list-style-type: none">2a. System wyświetla komunikat „Brak rezerwacji”3a. Pracownik recepcji rezygnuje z anulowania rezerwacji3b. Jeśli podczas rezerwacji podany został adres e-mail, pracownik może wysłać do klienta pocztą elektroniczną informację o anulowaniu rezerwacji
Specjalne wymagania:	<ol style="list-style-type: none">1. Wysoka niezawodność systemu2. Czas przetwarzania operacji anulowania rezerwacji nie może przekroczyć 5 sekund
Notatki i kwestie:	Brak

Tworzenie diagramu przypadku użycia

1. Identyfikacja przypadków użycia (np. z dokumentu wizji)
 - a) Wypisz je i otocz kółkiem tworząc use-case
 2. Identyfikacja aktorów
 3. Opracowanie związków (szczególnie asocjacji)
 - a) Popraw nazewnictwo
 4. Wykorzystanie kategorii zaawansowanych
 5. Udokumentowanie przypadku użycia
-

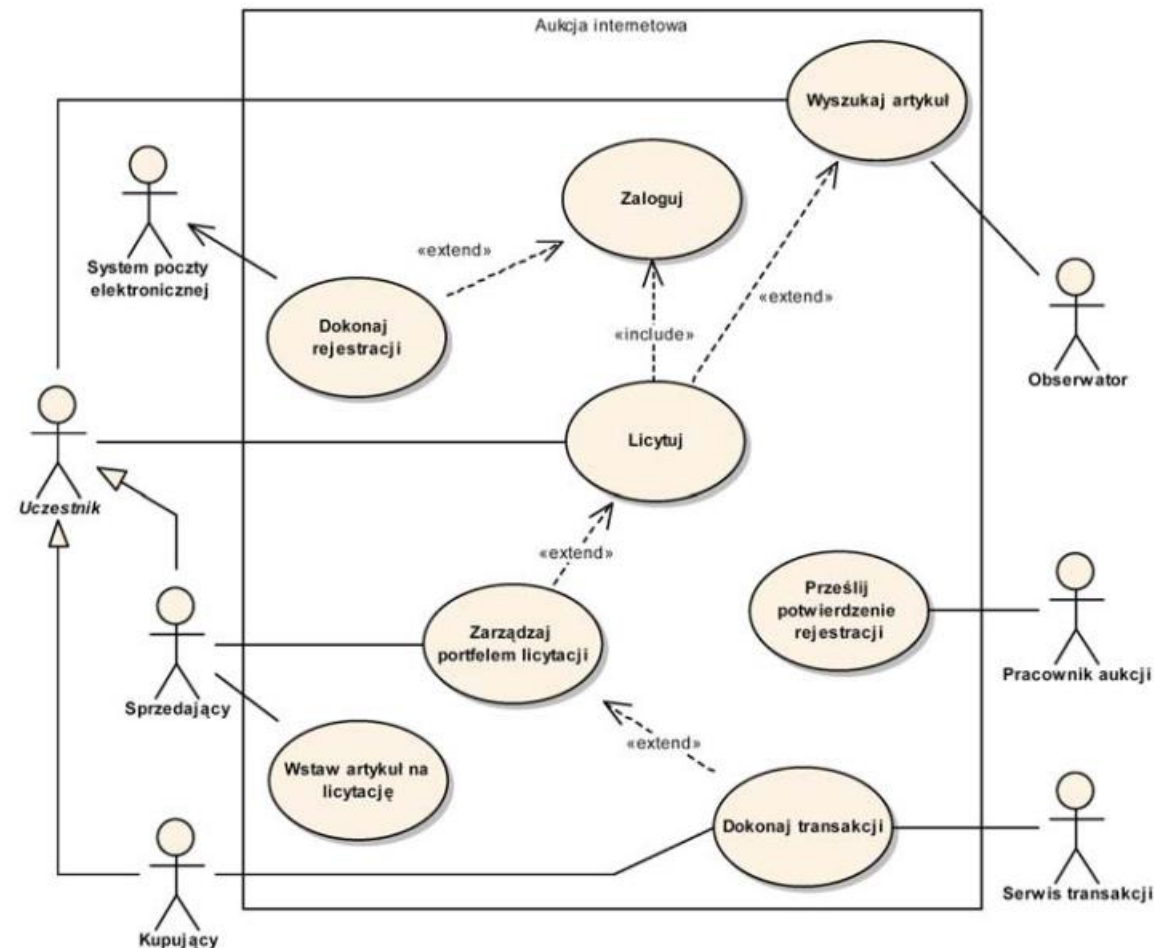
Przykład końcowy

- Aukcja internetowa
- Aby licytacja się odbyła zarówno *sprzedający* jak i *artykuł* muszą być zarejestrowane w systemie
- Podczas licytacji system aukcji zatwierdza kwoty proponowane przez *Kupujących*. Jeśli artykuł znajdzie nabywcę, *Kupujący* ustala z nim szczegóły płatności oraz przesyłki. W przeciwnym razie *Sprzedający* może usunąć artykuł z bazy systemu lub wystawić artykuł na nowej licytacji



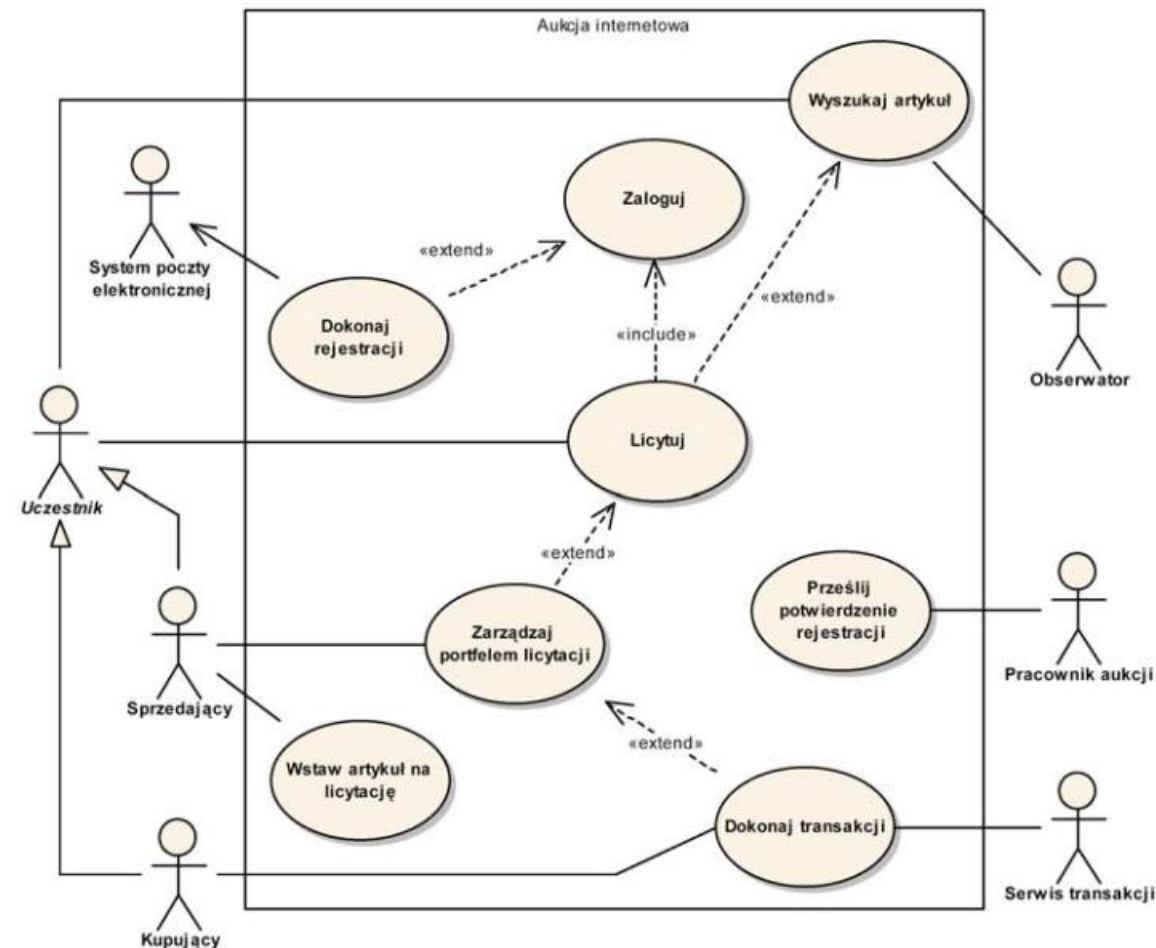
Przykład końcowy

- Aktorzy *Kupujący i Sprzedający* są uszczegółowieniami abstrakcyjnego aktora *Uczestnik*
- Oprócz wymienionych, na diagramie przypadków użycia systemu aukcji internetowej można wyszczególnić czterech innych aktorów:
- Serwis transakcji – zewnętrzny system informatyczny obsługujący transakcje
- Pracownik aukcji – potwierdza rejestrację Sprzedających



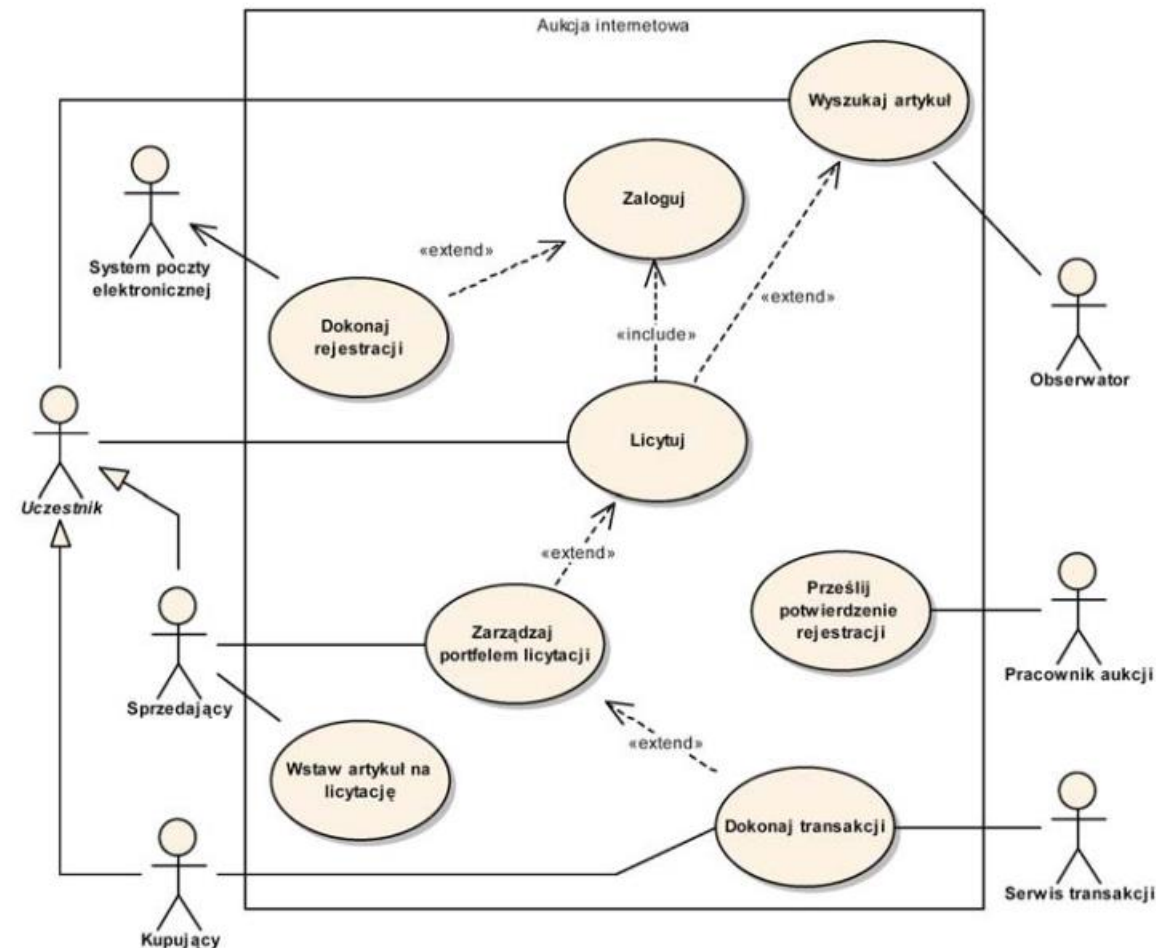
Przykład końcowy

- Obserwator, może wyłącznie wyszukać artykuły w katalogu aukcji
- System poczty elektronicznej odpowiada za wysłanie potwierdzenia rejestracji



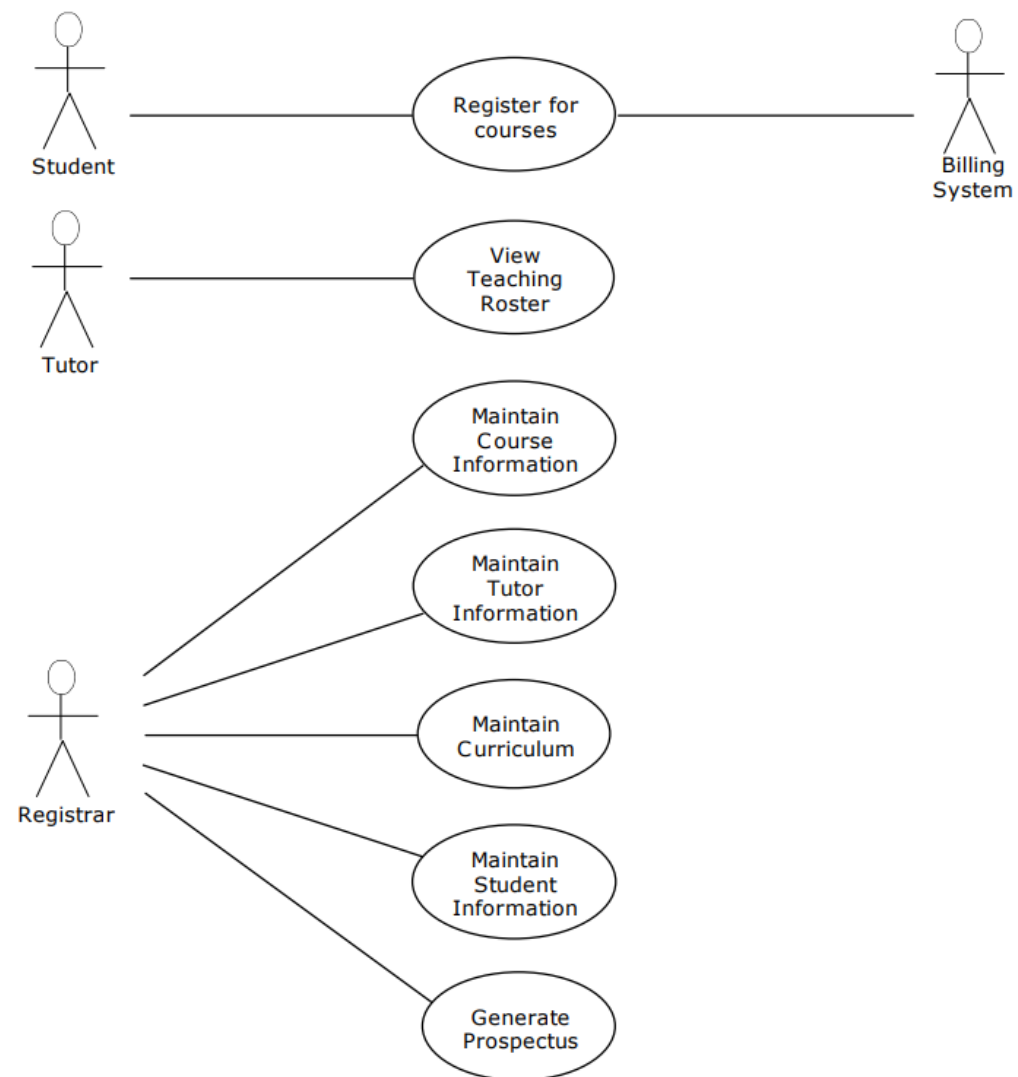
Przykład końcowy

- Sprzedający może korzystać z funkcjonalności aż 7 przypadków użycia
- Kupujący z 5
- Jakich ??



Przykład – Płatna szkoła

- Na początku semestru administracja tworzy listę kursów. Na liście kursów zawarte są takie informacje jak wykładowca, wydział oraz cele przedmiotu
 - W systemie student może utworzyć terminarz zajęć i wybrać z niego cztery kursy. Dodatkowo może wybrać dwa kursy, w przypadku jeżeli pierwszy wybór okaże się niedostępny. Żaden kurs nie może mieć więcej niż 10 uczestników. Żaden kurs nie może mieć więcej niż 3 uczestników. Funkcjonalność będzie dostępna dla wszystkich wewnętrznych użytkowników systemu.
 - System rozliczeniowy tworzy rachunek za wybrane kursy
 - Wykładowca używa systemu aby zidentyfikować kursy które prowadzi oraz studentów zapisanych na kursy
-



Przypadek użycia „Register for courses”

- Warunki wstępne: system pobiera od studenta numer albumu i hasło, wyświetla okno z dostępnymi kursami
 - Główny przepływ:
 - Stwórz terminarz
 - Potwierdź terminarz
 - Modyfikuj terminarz dodając lub usuwając kurs
 - Przepływ alternatywny
 - Jeżeli student wprowadzi niewłaściwy numer albumu, wyświetla błąd
 - Jeżeli terminarz istnieje, student zostanie spytany czy chce stworzyć nowy
 - Warunki końcowe: Student akceptuje terminarz, terminarz zostaje drukowany, system wysła dane do systemu rozliczeniowego
-



Opracowano na podstawie: „Język UML 2.0 w modelowaniu systemów informatycznych”
