

## Laboratorium 2

### **Specyfikacja wymagań funkcjonalnych przedstawiona za pomocą diagramu przypadków użycia**

Spis treści	
Cel laboratorium .....	2
UML.....	2
Diagram przypadków użycia (ang. Use Case Diagram, UCD).....	3
Przykład diagramu przypadków użycia.....	3
Scenariusz przypadku użycia .....	4

### Cel laboratorium

Na podstawie specyfikacji wymagań, zdefiniowanych w ramach poprzednich zajęć laboratoryjnych na leży zaprojektować diagram przypadków użycia oraz zaproponować scenariusze przypadków użycia..

### UML

Unified Modeling Language, czyli Zunifikowany Język Modelowania, to standardowy język graficzny używany do wizualizacji, specyfikacji, konstrukcji oraz dokumentacji systemów złożonych, głównie w zakresie inżynierii oprogramowania. UML jest wszechstronny i służy zarówno do modelowania struktury, jak i zachowań systemów.

### Główne cechy UML

- Standaryzacja – UML jest międzynarodowym standardem opracowanym przez organizację OMG (Object Management Group).
- Uniwersalność – wspiera modelowanie zarówno systemów oprogramowania, jak i procesów biznesowych.
- Obsługa różnych perspektyw – pozwala na tworzenie wielu rodzajów diagramów, które przedstawiają system z różnych punktów widzenia.

### Rodzaje diagramów UML

1. Diagramy strukturalne – opisują statyczną strukturę systemu:
  - Diagram klas – przedstawia klasy i ich zależności.
  - Diagram obiektów – opisuje instancje klas i ich relacje w określonym momencie.
  - Diagram komponentów – pokazuje zależności między komponentami oprogramowania.
  - Diagram wdrożenia – obrazuje fizyczne rozmieszczenie systemu na sprzęcie.
2. Diagramy behawioralne – opisują zachowanie systemu w czasie:
  - Diagram przypadków użycia – prezentuje funkcje systemu z perspektywy użytkowników.
  - Diagram aktywności – obrazuje przepływ działań lub procesów w systemie.
  - Diagram sekwencji – przedstawia interakcje między obiektami w ramach określonego scenariusza.
  - Diagram maszyny stanów – opisuje możliwe stany obiektu i przejścia między nimi.

### Zalety UML:

- Współpraca w zespole – umożliwia tworzenie spójnych, zrozumiałych modeli, które ułatwiają współpracę między członkami zespołu.
- Planowanie i dokumentacja – pomaga w analizie i projektowaniu systemu, a także w tworzeniu dokumentacji, która jest pomocna podczas dalszego rozwoju i konserwacji.
- Uniwersalność – można go używać zarówno do małych projektów, jak i złożonych systemów.

UML jest szeroko stosowany w branży IT, a jego znajomość jest często kluczowa dla analityków systemowych, architektów oprogramowania, projektantów i programistów.

### Diagram przypadków użycia (ang. Use Case Diagram, UCD)

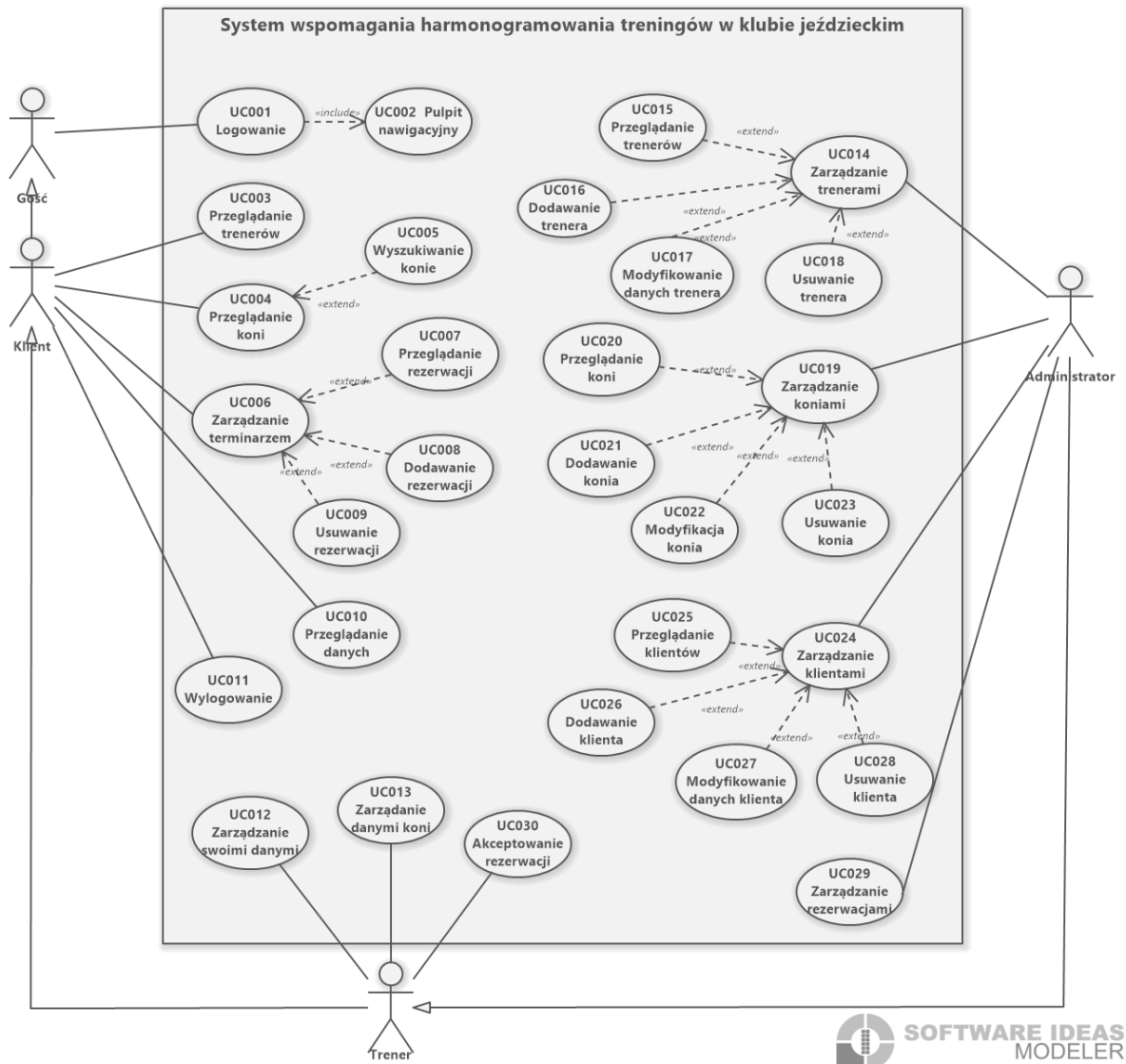
Narzędzie stosowane w inżynierii oprogramowania, szczególnie w analizie systemów, aby zilustrować funkcjonalność systemu i jego interakcje z użytkownikami lub innymi systemami. Składa się z kilku podstawowych elementów:

1. **Aktorzy** – przedstawiają użytkowników (lub inne systemy) wchodzących w interakcję z systemem. Mogą to być konkretne role użytkowników, np. Administrator, Klient lub Użytkownik.
2. **Przypadki użycia** – reprezentują funkcjonalności lub działania, które system oferuje aktorom. Przykłady to Zaloguj się, Wyślij wiadomość lub Złóż zamówienie. Każdy przypadek użycia jest reprezentowany przez owal.
3. **Relacje między przypadkami użycia:**
  - **Include** – relacja wskazująca, że dany przypadek użycia zawsze korzysta z innego przypadku użycia, np. Złóż zamówienie może zawsze korzystać z Wybierz produkt.
  - **Extend** – relacja wskazująca, że dany przypadek użycia może zostać rozszerzony przez inny, jeśli spełnione zostaną pewne warunki.
  - **Generalizacja** – relacja hierarchii między przypadkami użycia lub aktorami, np. Użytkownik i Administrator mogą być podklasami Osoby.
4. **System** – obszar, który zaznacza granice systemu, w którym przypadki użycia są realizowane.

### Przykład diagramu przypadków użycia

W projektowanym systemie wyróżnia się czterech aktorów, którzy w zależności od uprawnień posiadają pewne wspólne funkcjonalności systemu. Na Rys. 1 przedstawiono diagram przypadków użycia z uwzględnieniem podziału poszczególnych przypadków użycia oraz przyporządkowania ich do określonych aktorów tj. Gość, Klient, Trener, Administrator.

- Gość – aktor, który może zalogować się do systemu. Aktor Gość musi posiadać konto założone przez administratora systemu.
- Klient – jest to zalogowany użytkownik, ma wgląd w swoje dane, może przeglądać i wyszukiwać trenerów i konie, może tworzyć nowe rezerwacje oraz je anulować. Po-nadto może przeglądać swoje dane i wylogować się z systemu.
- Trener – aktor, który podobnie jak klient ma jedynie kilka funkcjonalności, dodatkowo może akceptować rezerwacje, które są przypisane do niego. Dodatkowo aktor ten może przeglądać wszystkich klientów w aplikacji oraz edytować dane konia.
- Administrator – aktor, który ma wszystkie uprawnienia. Ma dostęp do praktycznie wszystkich funkcjonalności, może edytować, dodawać oraz usuwać wszystkie rekordy takie jak konie, użytkownicy itp. Aktor ten ma wgląd we wszystkie rezerwacje w systemie może je anulować lub dodać nową.



Rys. 1. Diagram przypadków użycia projektowanego systemu.

## Scenariusz przypadku użycia

Scenariusz przypadku użycia to szczegółowy opis konkretnej ścieżki działania w ramach przypadku użycia, który przedstawia krok po kroku, jak aktor (użytkownik lub inny system) wchodzi w interakcję z systemem, aby osiągnąć określony cel. Scenariusze są przydatne do zrozumienia, jak system ma funkcjonować w praktyce, i pomagają w wykrywaniu potencjalnych wyjątków i warunków brzegowych.

## Struktura scenariusza przypadku użycia

Scenariusz przypadku użycia zazwyczaj zawiera:

- Nazwa scenariusza – jasne określenie, czego dotyczy dany scenariusz.
- Aktor – osoba lub system, który inicjuje interakcję.
- Cel – co aktor chce osiągnąć, korzystając z funkcji systemu.
- Warunki początkowe (przesłanki) – sytuacje lub stany systemu, które muszą być spełnione przed rozpoczęciem scenariusza.

## Inżynieria Oprogramowania – Laboratorium

- Przebieg główny (scenariusz podstawowy) – opis kroków interakcji w idealnym przypadku, kiedy wszystko przebiega pomyślnie.
- Scenariusze alternatywne – kroki, które mogą zostać wykonane w zależności od różnych sytuacji lub warunków (np. jeśli aktor wprowadzi niepoprawne dane).
- Warunki końcowe – stan systemu po zakończeniu scenariusza, jeśli zakończył się sukcesem lub niepowodzeniem.

### Przykład scenariusza przypadku użycia

Przypadek użycia: Zaloguj się do systemu

- Aktor: Użytkownik
- Cel: Użytkownik chce uzyskać dostęp do swojego konta.
- Warunki początkowe: Użytkownik jest zarejestrowany i zna swoje dane logowania.
- Przebieg główny:
  1. Użytkownik otwiera stronę logowania.
  2. Użytkownik wprowadza swoją nazwę użytkownika i hasło.
  3. System weryfikuje dane logowania.
  4. System przyznaje dostęp i przekierowuje użytkownika do strony głównej.
- Scenariusz alternatywny (dla kroku 3):
  - Jeśli dane logowania są niepoprawne, system wyświetla komunikat o błędzie i pozwala na ponowne wprowadzenie danych.
- Warunki końcowe: Użytkownik jest zalogowany i ma dostęp do swojego konta albo widzi komunikat o błędnych danych logowania.

#### Scenariusz 1 Logowanie do systemu

Element	Opis
<b>Przypadek użycia</b>	Zaloguj się do systemu (UC001)
<b>Aktor</b>	Użytkownik
<b>Cel</b>	Użytkownik chce uzyskać dostęp do swojego konta
<b>Warunki początkowe</b>	Użytkownik jest zarejestrowany i zna swoje dane logowania
<b>Przebieg główny</b>	1. Użytkownik otwiera stronę logowania. 2. Użytkownik wprowadza swoją nazwę użytkownika i hasło. 3. System weryfikuje dane logowania. 4. System przyznaje dostęp i przekierowuje użytkownika do strony głównej.
<b>Scenariusz alternatywny</b>	Krok 3: Jeśli dane logowania są niepoprawne, system wyświetla komunikat o błędzie i pozwala na ponowne wprowadzenie danych.
<b>Warunki końcowe</b>	Użytkownik jest zalogowany i ma dostęp do swojego konta albo widzi komunikat o błędnych danych logowania.

Scenariusze przypadków użycia stanowią dokładne odwzorowanie interakcji między aktorem a systemem, co ułatwia ich implementację i testowanie.