**Projekt systemu Komis Samochodowy**

Podstawy Inżynierii Oprogramowania

**Spis treści:**

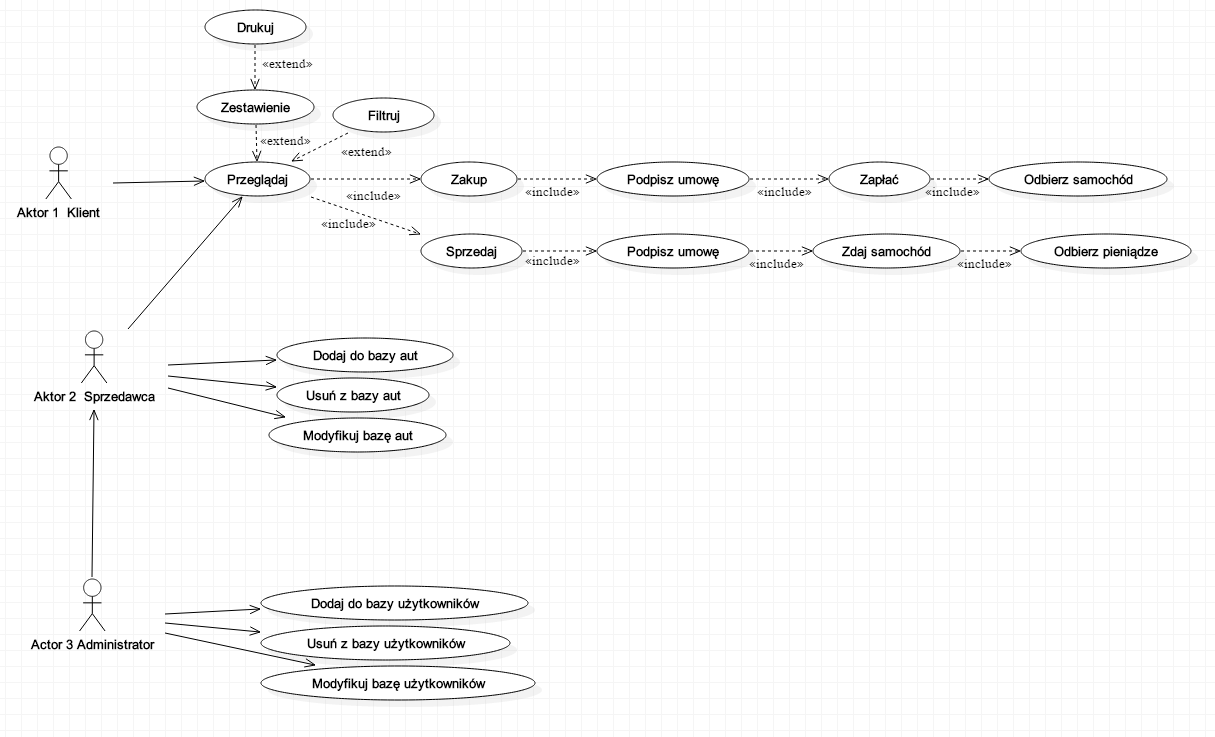
1. Opis funkcjonalności systemu.
2. Diagram przypadków użycia.
3. Diagram klas.
4. Diagram obiektów.
5. Diagram pakietów.
6. Diagram aktywności.
7. Diagram komunikacji.
8. Podsumowanie.
9. **Opis funkcjonalności systemu.**

System oparty jest na umowie komisu, obsługujący klientów prywatnych. Klient decyduje się na kupno lub sprzedaż samochodu na rzecz komisu. W ramach podpisywanej umowy klient zobowiązuje się dodatkowo uiścić opłatę prowizyjną za wykonywaną usługę.

Aktor 1 - wyżej wymieniony klient - będzie w stanie przeglądać interesujące go oferty. W ramach przyśpieszenia tego procesu będzie miał do dyspozycji narzędzie filtrujące, które pozwoli mu ograniczyć ilość ofert. Klient będzie mógł również stworzyć własną, spersonalizowaną listę ofert z możliwością wydrukowania jej podsumowania. Ostatnią opcją, którą będzie mógł wykonać aktor 1. jest sprzedaż lub zakup interesujących go produktu/produktów, co będzie się bezpośrednio wiązało z podpisaniem odpowiedniej umowy, odbiorem lub zdaniem samochodu (oraz wszystkich związanych z nim przedmiotów) oraz otrzymaniem bądź wpłaceniem odpowiedniej kwoty na rzecz komisu samochodowego.

Aktor 2 – wyżej wymieniony sprzedawca – jest przede wszystkim odpowiedzialny za dodawanie, usuwanie oraz modyfikowanie rekordów bazy danych komisu. Równie ważnym zadaniem będzie obsługa klienta, przez co rozumiemy: nadzorowanie procesu sprzedaży bądź kupna, przygotowanie specjalnej umowy ,zaksięgowanie wpłaty lub wydanie pieniędzy i odpowiednio wydanie bądź przyjęcie samochodu.

1. **Diagram przypadków użycia.**



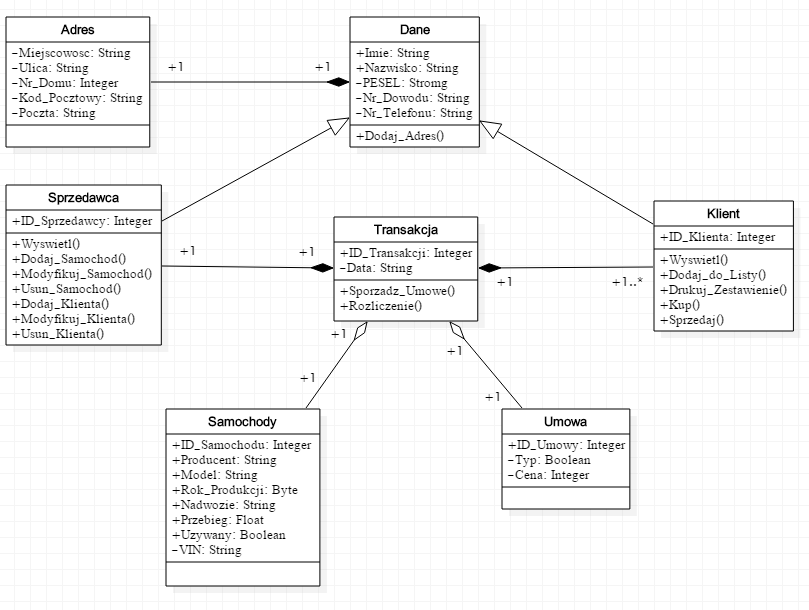
* 1. Definicja i zastosowanie

Diagram przypadków użycia (ang. use case diagram) jest diagramem, który przedstawia funkcjonalność systemu wraz z jego otoczeniem.  
Diagramy przypadków użycia pozwalają na graficzne zaprezentowanie własności systemu tak, jak są one widziane po stronie użytkownika.  
Diagramy przypadków użycia służą do zobrazowania usług, które są widoczne z zewnątrz systemu.

* 1. Opis

Nasz diagram prezentuje różne przypadki użycia w każdej sytuacji, jaką przewiduje system. Wydzielone posiadamy trzy poziomy uprawnień, które organizują nam hierarchię w systemie. Dzięki temu łatwo da się zauważyć, jaka funkcja należy do jakiego użytkownika.

1. **Diagram klas**.



* 1. Definicja i zastosowanie

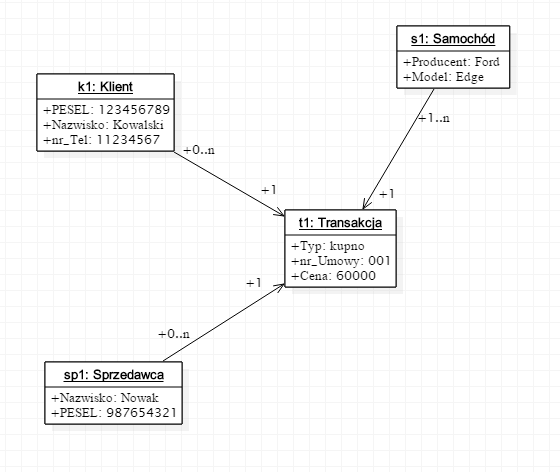
Diagram klas obrazuje pewien zbiór klas, interfejsów i kooperacji oraz związki między nimi. Jest on grafem złożonym z wierzchołków (klas, interfejsów, kooperacji) i łuków (reprezentowanych przez relacje). Diagram klas stanowi opis statyki systemu, który uwypukla związki między klasami, pomijając pozostałe charakterystyki. Najsilniej prezentuje on więc strukturę systemu, stanowiąc podstawę dla jego konstrukcji. W modelowaniu złożonych systemów nie mamy obowiązku przedstawiania ich struktury na jednym diagramie. Powinniśmy pamiętać o tym, że złożenie wszystkich diagramów, a właściwie ich elementów  
i relacji, stanowi kompletny model. Możemy zatem przyjąć, że podzbiory zbioru klas użyte na diagramach klas są wybierane celowo i stanowią wynik decyzji zarówno analitycznych, jak i projektowych. Tak na przykład, diagramy klas stanowiące wynik decyzji analitycznych tworzą tzw. widoki klas biorących udział w realizacji danego przypadku użycia i powiązania między nimi.

Zasadniczo jednak diagramy klas służą do zobrazowania statycznych aspektów perspektywy projektowej, w której bierze się pod uwagę wymagania funkcjonalne systemu – usługi, jakie system powinien udostępniać swoim użytkownikom.

* 1. Opis

Diagram pomaga nam określić, jakie należy stworzyć w systemie klasy. Klasa „Sprzedawca” oraz „Klient” korzysta z klasy „Dane”. Klasa „Transakcja” korzysta i zależy od klas „Sprzedawcy” i” Klienta” by jej metody były kompletne. Tak samo jest z klasą „Dane” oraz „Adres”. Klasy „Samochody” oraz „Umowa” współdzielą odwołanie do klasy „Transakcja”. Każdy opisany związek z klasami (prócz dziedziczenia) ma określoną krotność, określającą ilość obiektów danych klas.

1. **Diagram obiektów**.



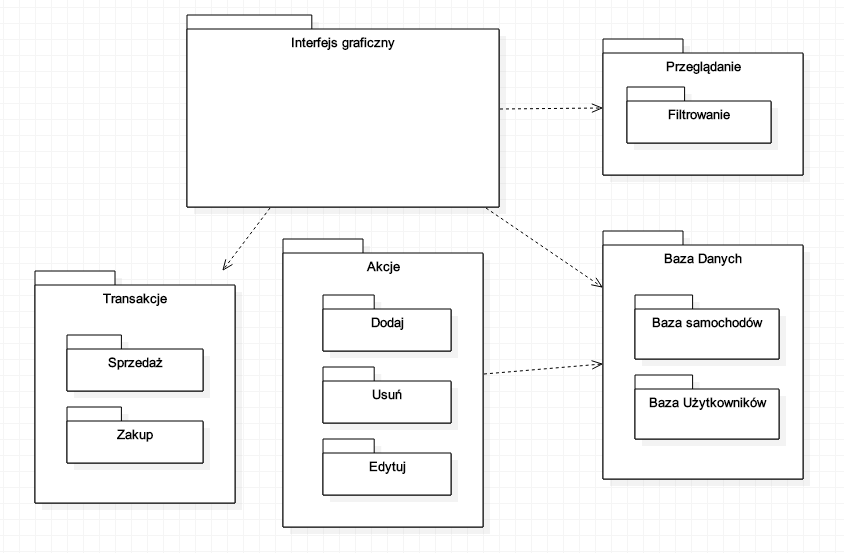
* 1. Definicja i zastosowanie

Podobnie do diagramu klas, przedstawiają jednak nie klasy, tylko konkretne obiekty będące instancjami klas systemu. Z punktu widzenia notacji diagramy obiektów używaj ą elementów zapożyczonych z diagramów klas, chociaż często używaj ą prostszej notacji. Skupiają się na obiektach a nie na związkach pomiędzy klasami. Większość z nich u żywa wyłącznie obiektów i asocjacji. Diagram jest więc wizualizacją hipotetycznego stanu systemu podczas jego działania. Służy do tworzenia przykładów pomagających zrozumieć diagram klas, a przede wszystkim powiązań w nim występujących.

* 1. Opis

Na tym diagramie widać przykłady obiektów klas opisanych powyżej. Obiekty „k1”, „s1” oraz „sp1” odnoszą się do obiektu „t1”, gdyż na tej podstawie sporządzana jest umowa kupna \ sprzedaży. Tak więc, by powstał jeden obiekt klasy „Transakcja”, potrzebny jest obiekt z klas „Klient”, „Samochód” i „Sprzedawca”.

1. **Diagram pakietów**.



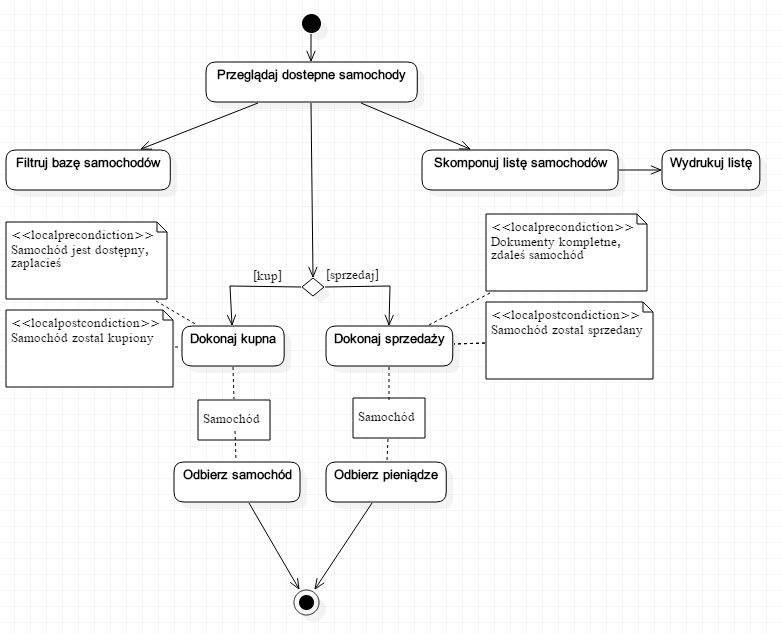
* 1. Definicja i zastosowanie

Diagram pakietów (ang. package diagram) jest strukturalnym diagramem, który prezentuje pakiety i relacje zachodzące pomiędzy nimi. Diagram pakietów służy do modelowania agregatów bytów, jakimi są pakiety. Diagramy pakietów pozwalają na modelowanie systemu na wysokim stopniu abstrakcji, gdyż pakiety reprezentują ogromną liczbę klas, interfejsów, diagramów i innych bytów – pozwala to wyeksponować tylko zasadnicze funkcje systemu.

* 1. Opis

Wyróżnione zostały tutaj istotne zadania systemu, które wykorzystywane będą w największym stopniu.

1. **Diagram aktywności**.



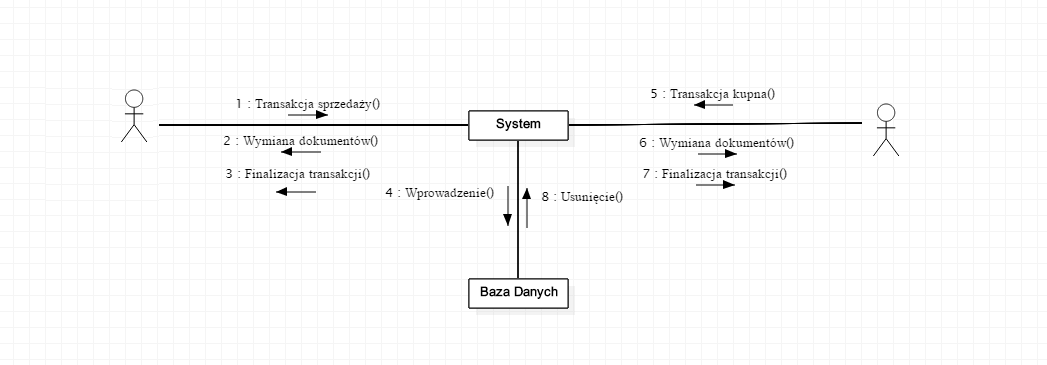
* 1. Definicja i zastosowanie

Diagram czynności (ang. activity diagram) jest diagramem interakcji, który służy do modelowania dynamicznych aspektów systemu. Jego zasadniczą funkcją jest przedstawienie sekwencji kroków, które są wykonywane przez modelowany fragment systemu. Diagram sekwencji pozwala także na prezentację przepływów współbieżnych oraz na zaprezentowanie zmian stanów obiektów podczas przechodzenia pomiędzy czynnościami.  Definiując ogólnie, diagram czynności jest stosowany do modelowania behawioralnych aspektów systemu i obrazuje strumień kolejno wykonywanych czynności (ang. fast track).

* 1. Opis

Diagram ten przedstawia sposób krok po kroku jak przebiega proces korzystania z systemu. Uwzględniona została każda opcja, jaka dostępna jest do wykorzystania dla użytkownika w systemie. Pod każdą akcją kryją się nawiązania do obiektów w poprzedniego diagramu.

1. **Diagram komunikacji**.



* 1. Definicja i zastosowanie

Diagram komunikacji (ang. communication diagram) jest rozszerzoną i przemianowaną wersją diagramu współdziałania znanego z UML 1.x. Skupia się on na obiektach wchodzących w skład interakcji i wymienianymi przez nie komunikatach, natomiast w mniejszym stopniu niż diagram sekwencji (choć nadal obecnym) wskazuje na aspekt czasowy. Z tego powodu obiekty na diagramie komunikacji są umieszczone tak, aby łatwo można było opisać ich relacje pomiędzy sobą. Komunikacje są przedstawiane za pomocą linii łączących obiekty, natomiast przesyłane między obiektami komunikaty i dane są umieszczane obok tych linii. Każdy komunikat jest opatrzony etykietą liczbową, wskazującą na kolejność ich wysyłania.

* 1. Opis

Przedstawiony mamy powyżej schemat, obrazujący sposób przepływu informacji między użytkownikami systemu oraz samym systemem.

1. **Podsumowanie**.

Budowanie rozwiązań informatycznych poprzedza określenie potrzeb. Potrzeby te, zwane wymaganiami, pozwalają określić jaki ma być system, jakie funkcje ma realizować. Przy zastosowaniu w projekcie języka UML, mogą zostać również zapisane w formie graficznej. Możliwość odwzorowania graficznego wymagań funkcjonalnych stawianych systemowi, jest niewątpliwie jedną z największych zalet języka UML.

Udało się nam zaprojektować taki system, przy użyciu odpowiednich narzędzi oraz wiedzy. Przekonaliśmy się, że takie zadanie nie należy do łatwych, lecz stworzenie kompletnego projektu z pewnością pomaga w znalezieniu problemów, które wyniknęłyby podczas tworzenia takiej aplikacji, jak i dobrze przemyślany projekt równie dobrze rokuje na powodzenie i sukces naszej aplikacji.