

Wydział



Uniwersytetu Gdańskiego



Uniwersytet Gdański
Wydział Matematyki, Fizyki i Informatyki
Instytut Informatyki

Imię i Nazwisko (nr indeksu)	Agata Bartczak (285755), Sebastian Jabłoński (285810)
Nazwa uczelni	Uniwersytet Gdański
Kierunek	Informatyka (profil praktyczny)
Prowadzący	dr inż. Stanisław Witkowski
Specjalność	—
Nazwa ćwiczenia	Planowanie procesu wdrożenia produktu informatycznego za pomocą UML
Numer sprawozdania	4
Data zajęć	08.11.2024
Data oddania	14.11.2024
Miejsce na ocenę	

Spis treści

1	Elementy języka UML	2
2	Modelowanie wymagań za pomocą przypadków użycia	2
3	Diagramy czynności i sekwencji	2
4	Modelowanie klas i powiązań między nimi	3
5	Diagramy komponentów	3
6	Podział modelu na pakiety	4
7	Modelowanie wdrożenia systemu	4
8	Visual Paradigm online - opis działania	4
9	Wnioski	15

1 Elementy języka UML

Język UML (Unified Modeling Language) to uniwersalne narzędzie wykorzystywane do modelowania systemów informatycznych. Jego kluczowymi elementami są diagramy, które pozwalają na wizualizację różnych aspektów systemu, takich jak struktura, zachowanie i interakcje. Podstawowe elementy UML obejmują:

- Diagramy strukturalne (np. diagramy klas, obiektów, komponentów i wdrożenia),
- Diagramy behawioralne (np. przypadki użycia, diagramy czynności, maszyny stanowe),
- Diagramy interakcji (np. diagramy sekwencji, komunikacji).

Elementy te umożliwiają inżynierom oprogramowania lepsze zrozumienie wymagań, projektowanie architektury oraz dokumentację systemów.

2 Modelowanie wymagań za pomocą przypadków użycia

Modelowanie wymagań za pomocą przypadków użycia jest techniką pozwalającą na identyfikację i zrozumienie interakcji użytkowników z systemem. Diagramy przypadków użycia przedstawiają aktorów (rolę lub użytkowników) i związane z nimi funkcjonalności systemu. Każdy przypadek użycia opisuje sekwencję zdarzeń, które prowadzą do uzyskania konkretnego rezultatu z punktu widzenia użytkownika.

- Aktorzy – reprezentują role, które wchodzą w interakcje z systemem.
- Relacje – wskazują związki pomiędzy aktorami a przypadkami użycia (np. relacje typu «include» i «extend»).
- Scenariusze – szczegółowe opisy zachowań związanych z przypadkiem użycia.

Dzięki diagramom przypadków użycia analitycy mogą zdefiniować wymagania funkcjonalne systemu w sposób zrozumiały dla interesariuszy.

3 Diagramy czynności i sekwencji

Diagramy czynności oraz diagramy sekwencji to narzędzia do modelowania zachowań systemu:

- Diagramy czynności są używane do przedstawiania przepływu pracy lub procesów biznesowych. Pokazują, jak różne działania są ze sobą powiązane, oraz wskazują decyzje, równoległe zadania i synchronizacje.

- Diagramy sekwencji koncentrują się na kolejności komunikacji między obiektami lub komponentami. Wskazują, jakie wiadomości są przesyłane między obiektami oraz w jakiej kolejności. Diagramy te są przydatne do modelowania szczegółowych interakcji i logiki sterowania.

Te diagramy pomagają zrozumieć dynamikę systemu i są wykorzystywane na etapie analizy oraz projektowania.

4 Modelowanie klas i powiązań między nimi

Diagramy klas są podstawowym narzędziem do modelowania struktury systemu. Przedstawiają klasy, ich atrybuty, metody oraz powiązania między nimi:

- Klasy – reprezentują obiekty w systemie, posiadają atrybuty i operacje.
- Asocjacje – definiują relacje pomiędzy klasami, np. powiązania jedno- lub wielokrotne.
- Dziedziczenie – ilustruje hierarchię klas, umożliwiając dziedziczenie atrybutów i metod.
- Agregacje i kompozycje – pokazują relacje „całość-część”, przy czym kompozycja oznacza silne powiązanie.
- Diagramy klas pomagają w zrozumieniu struktury systemu i są podstawą do projektowania kodu.

5 Diagramy komponentów

Diagramy komponentów służą do wizualizacji modułowej struktury systemu. Każdy komponent reprezentuje odrębną jednostkę oprogramowania, która realizuje określoną funkcjonalność i może być rozwijana oraz wdrażana niezależnie:

- Komponenty – moduły oprogramowania, które mają jasno zdefiniowane interfejsy.
- Interfejsy – definiują punkty połączeń między komponentami.
- Relacje zależności – wskazują, jakie komponenty są ze sobą powiązane lub które komponenty wymagają innych do poprawnego działania.
- Diagramy te są istotne przy projektowaniu architektury systemów rozproszonych oraz dużych projektów, gdzie separacja funkcji jest kluczowa.

6 Podział modelu na pakiety

Diagramy pakietów pozwalają na organizację złożonych systemów w logiczne grupy, zwane pakietami. Ułatwiają zarządzanie dużymi projektami przez segregowanie klas, komponentów i diagramów:

- Pakiety – logiczne grupy elementów, które mogą zawierać klasy, interfejsy, podpakiety.
- Zależności między pakietami – wskazują, jak jeden pakiet wpływa na drugi, co pomaga w identyfikacji zależności między różnymi częściami systemu.

Podział na pakiety wspiera modularność i skalowalność systemu.

7 Modelowanie wdrożenia systemu

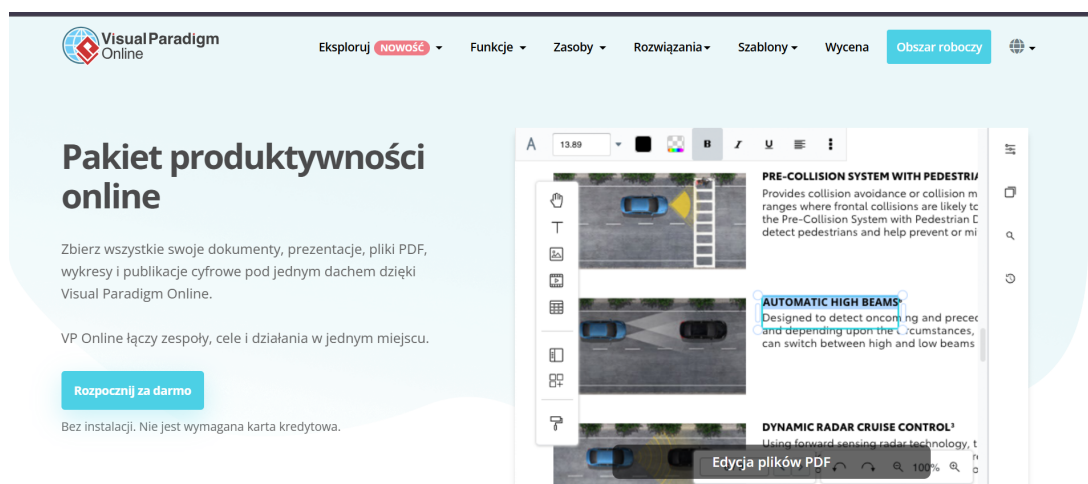
Diagramy wdrożenia są wykorzystywane do przedstawienia fizycznej architektury systemu, w tym sprzętu, na którym system jest uruchamiany:

- Węzły – reprezentują jednostki sprzętowe, np. serwery lub urządzenia sieciowe.
- Artefakty – pliki lub inne fizyczne jednostki wdrażane na węzłach.
- Powiązania – wskazują na komunikację pomiędzy węzłami.

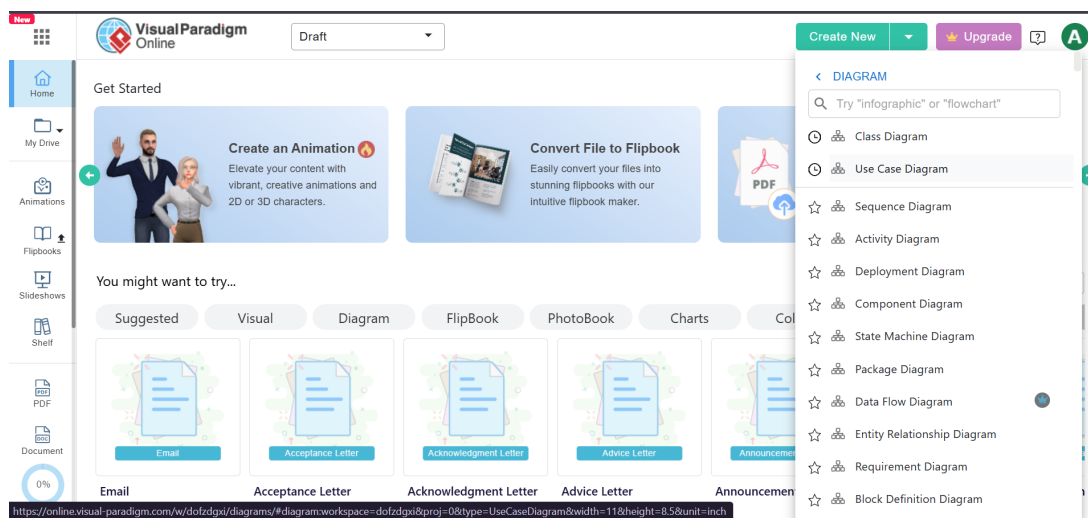
Diagramy wdrożenia są szczególnie przydatne do planowania infrastruktury oraz pokazania rozmieszczenia aplikacji na serwerach w kontekście sieciowym.

8 Visual Paradigm online - opis działania

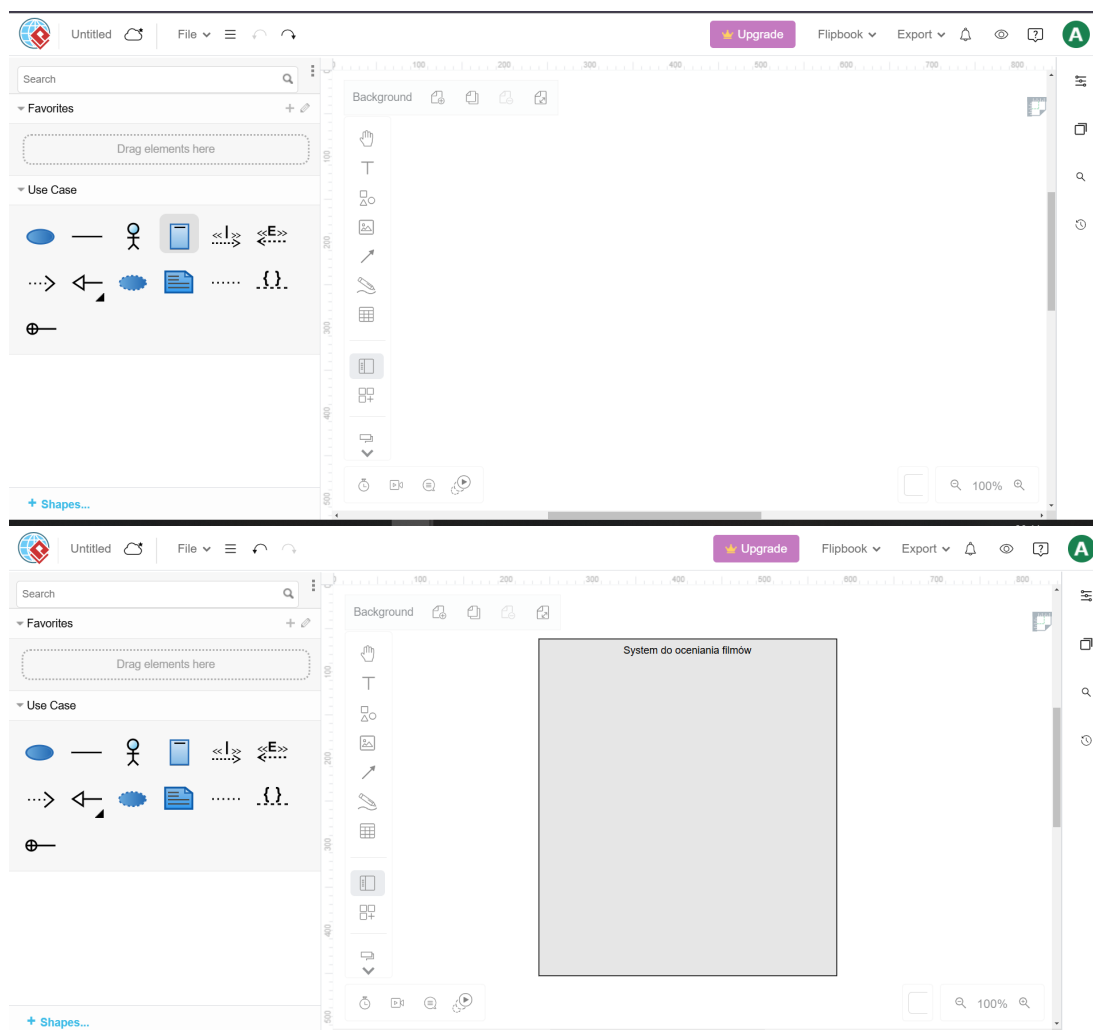
Visual Paradigm online to serwis, który pozwala m.in. na tworzenie diagramów UML. Chcąc uzyskać dostęp do funkcji Visual Paradigm online należy utworzyć konto i się zalogować (przycisk „Rozpocznij za darmo”).



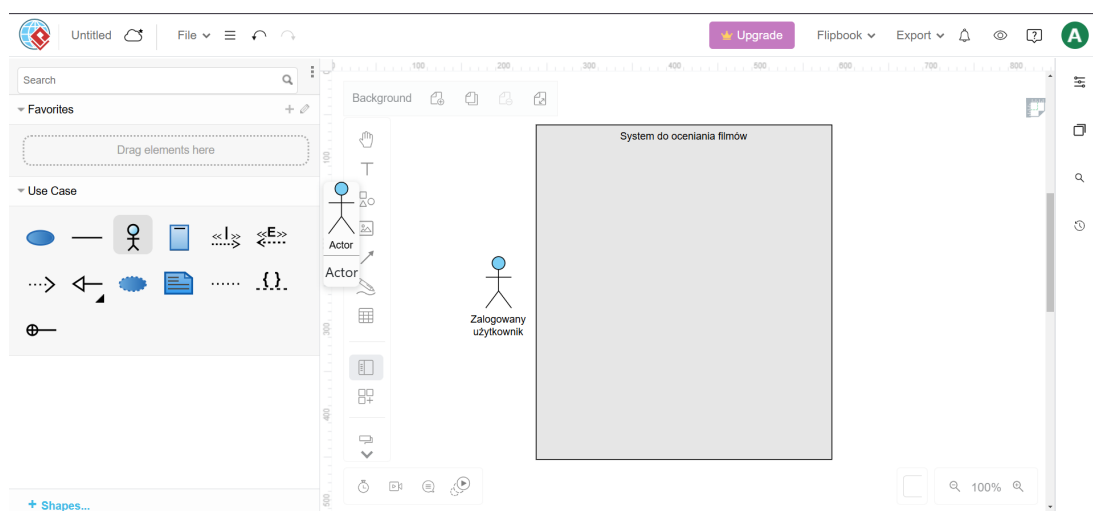
Aby utworzyć nowy diagram klikamy przycisk „Create New”, następnie „Diagram” i wybieramy pożądany template. Do utworzenia diagramu przypadku użycia wybrano „Use Case Diagram”



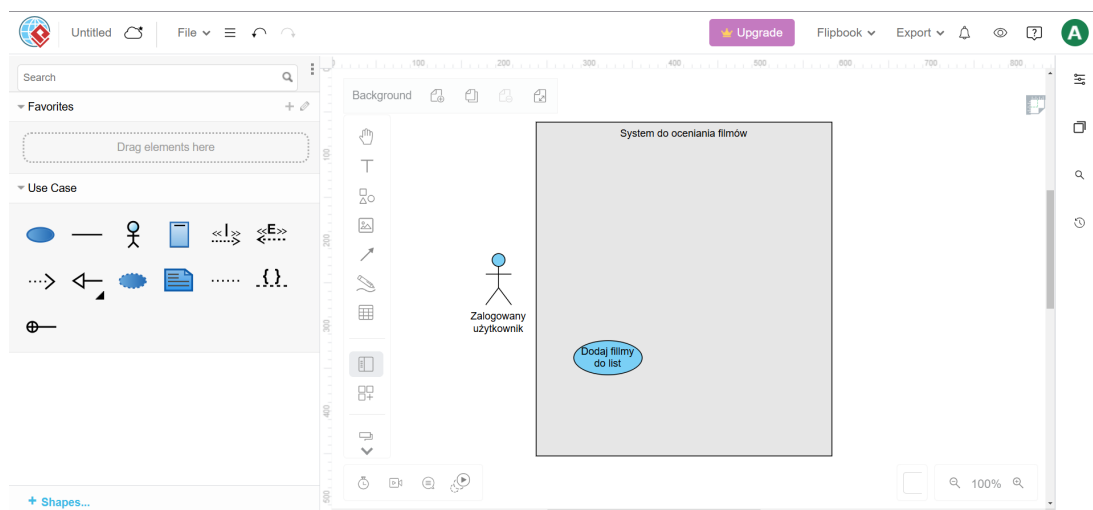
Dodanie nowego systemu. Wygląd poszczególnych elementów można zmieniać poprzez kliknięcie na element i wybranie odpowiednich opcji z paska wyżej. Zmieniono podpis systemu na „System do oceniania filmów” i zwiększono jego powierzchnię poprzez przeciągnięcie jednego z narożników.



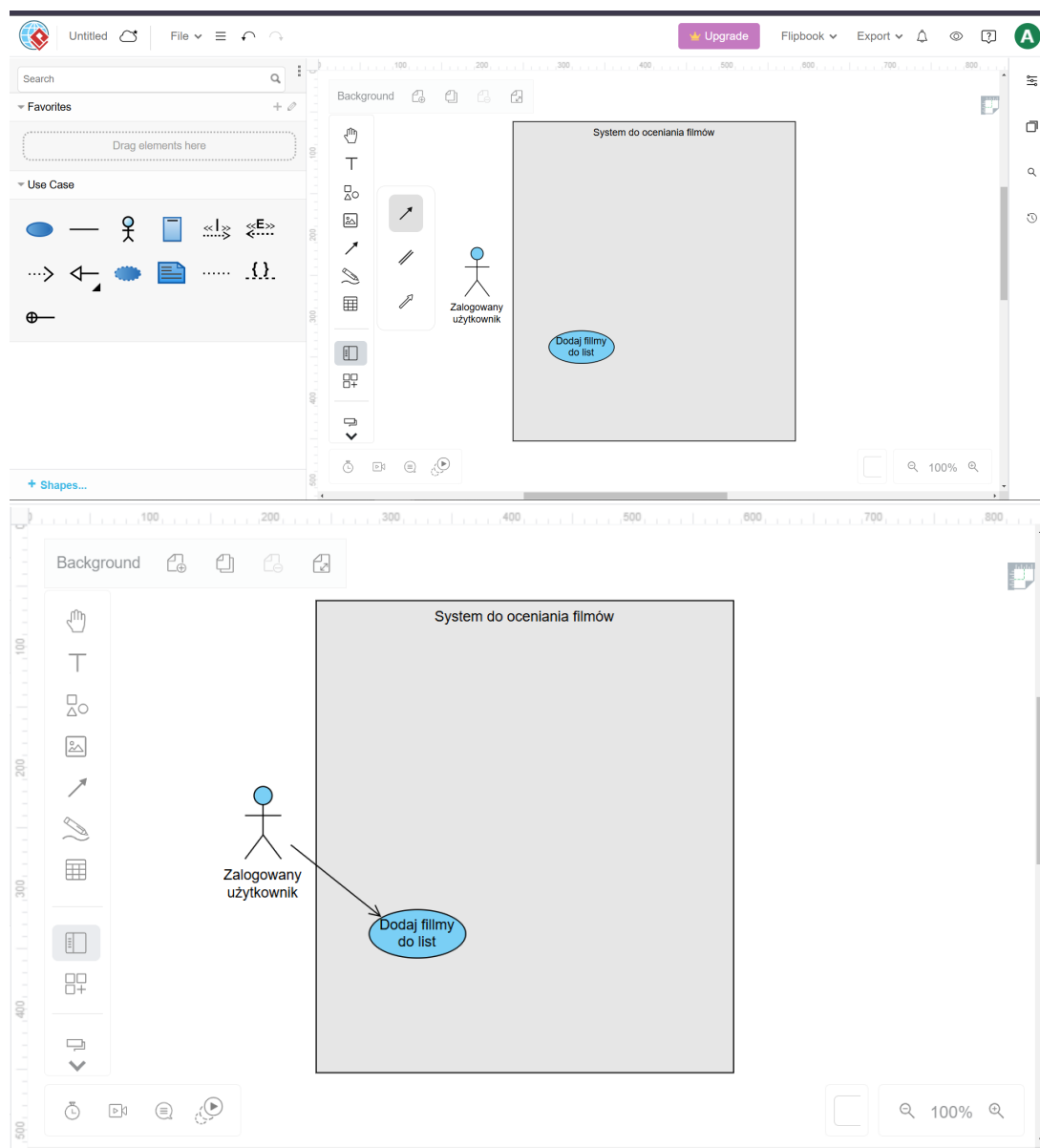
Dodawanie aktorów. Zmiana położenia następuje przez przegiąnięcie elementu po obszarze.



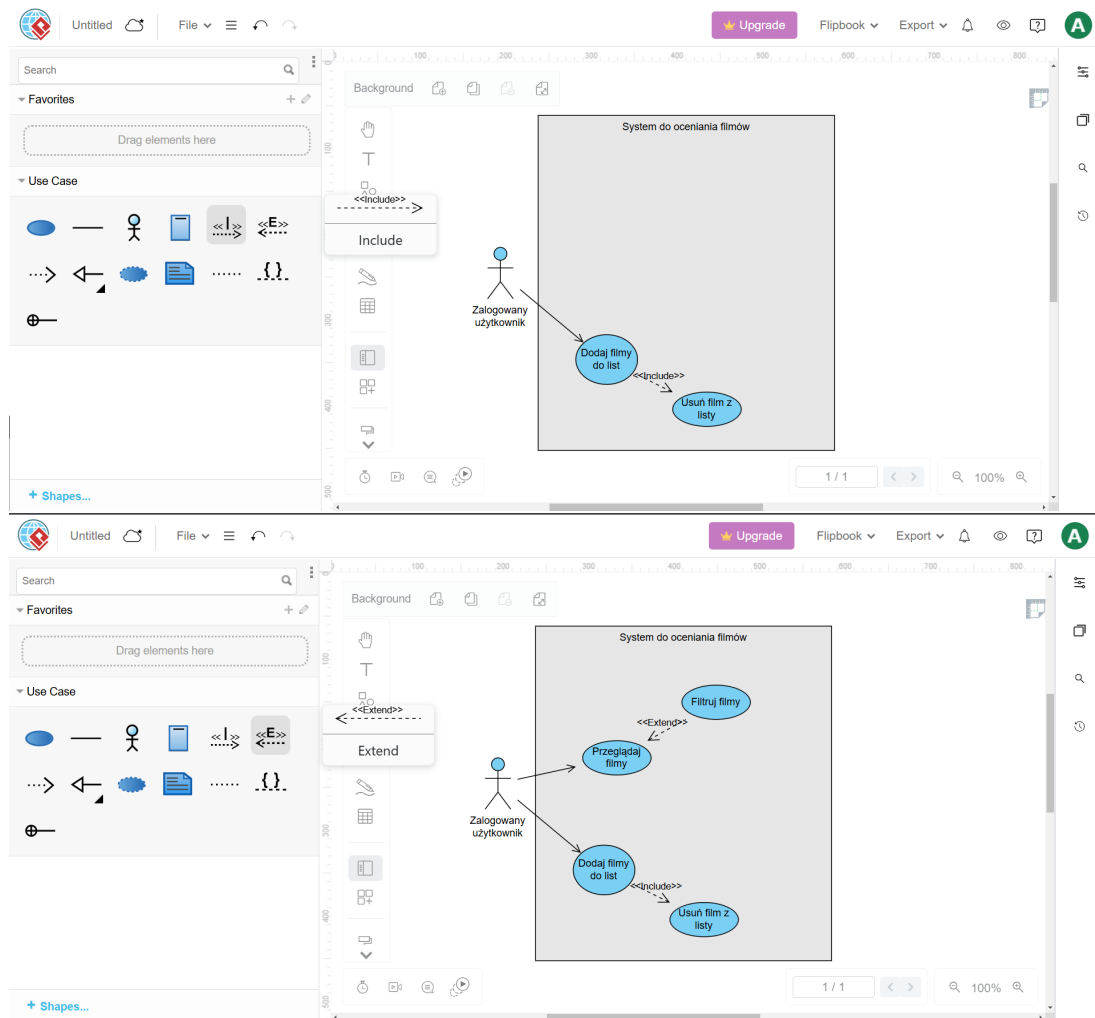
Dodawanie przypadku użycia (ikona elipsy z ciągłym obramowaniem).



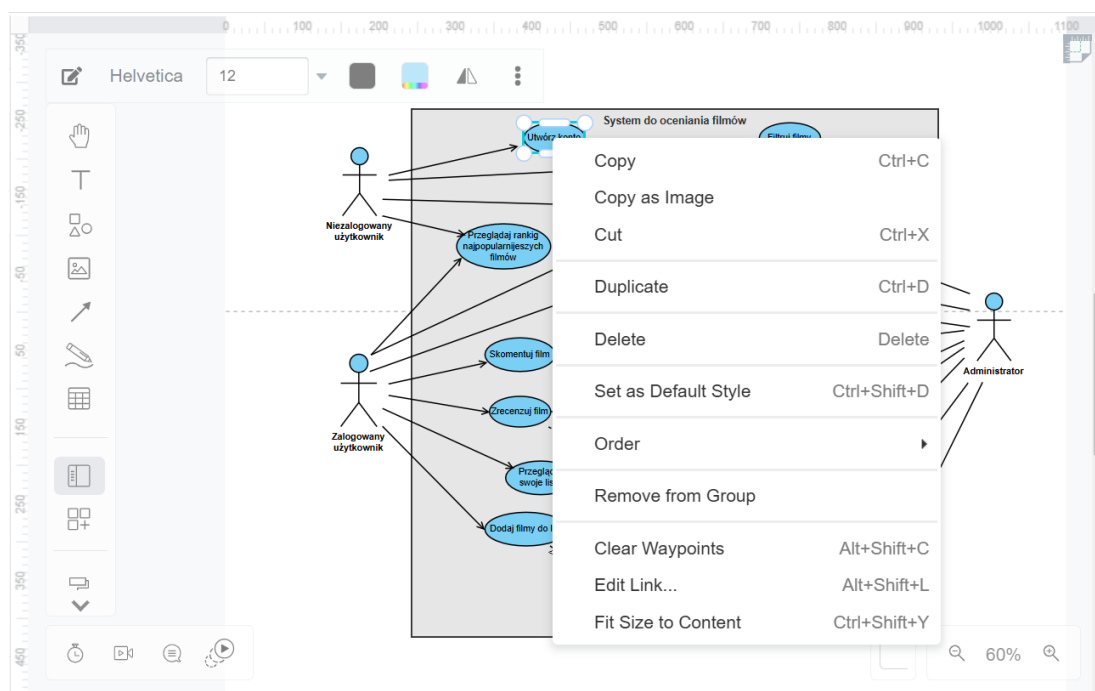
Dodawanie relacji. Po wybraniu odpowiedniego narzędzia z bocznego panelu (ikona strzałki) należy przeciągnąć linię od elementu, z którego wychodzi relacja do punktu, do którego wchodzi relacja.



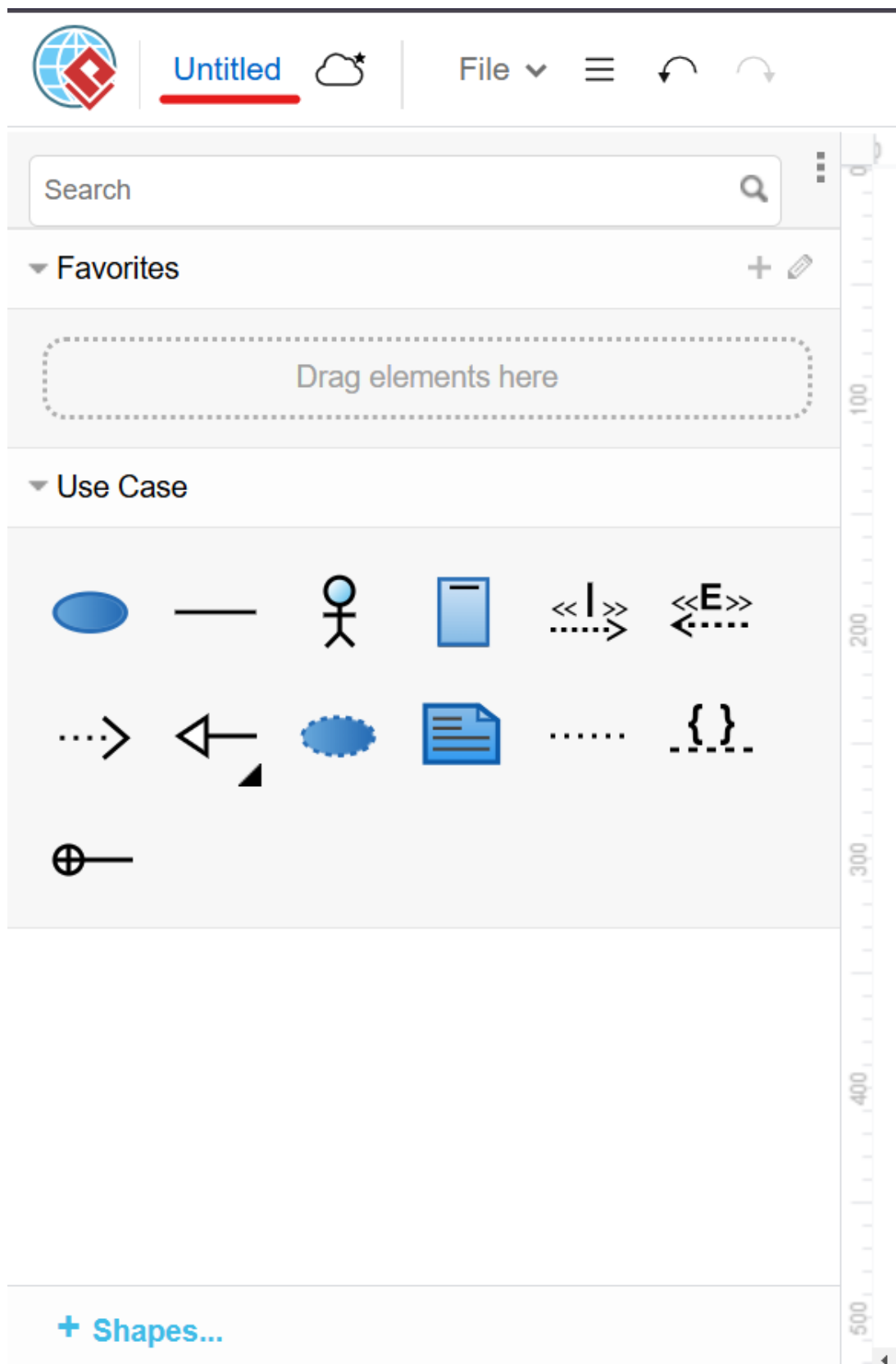
Aby utworzyć relację „include” lub „extends” należy wybrać odpowiednie narzędzie (ikona z literą „I” lub literą z „E”) i przeciągnąć od punktu początkowego do końcowego.



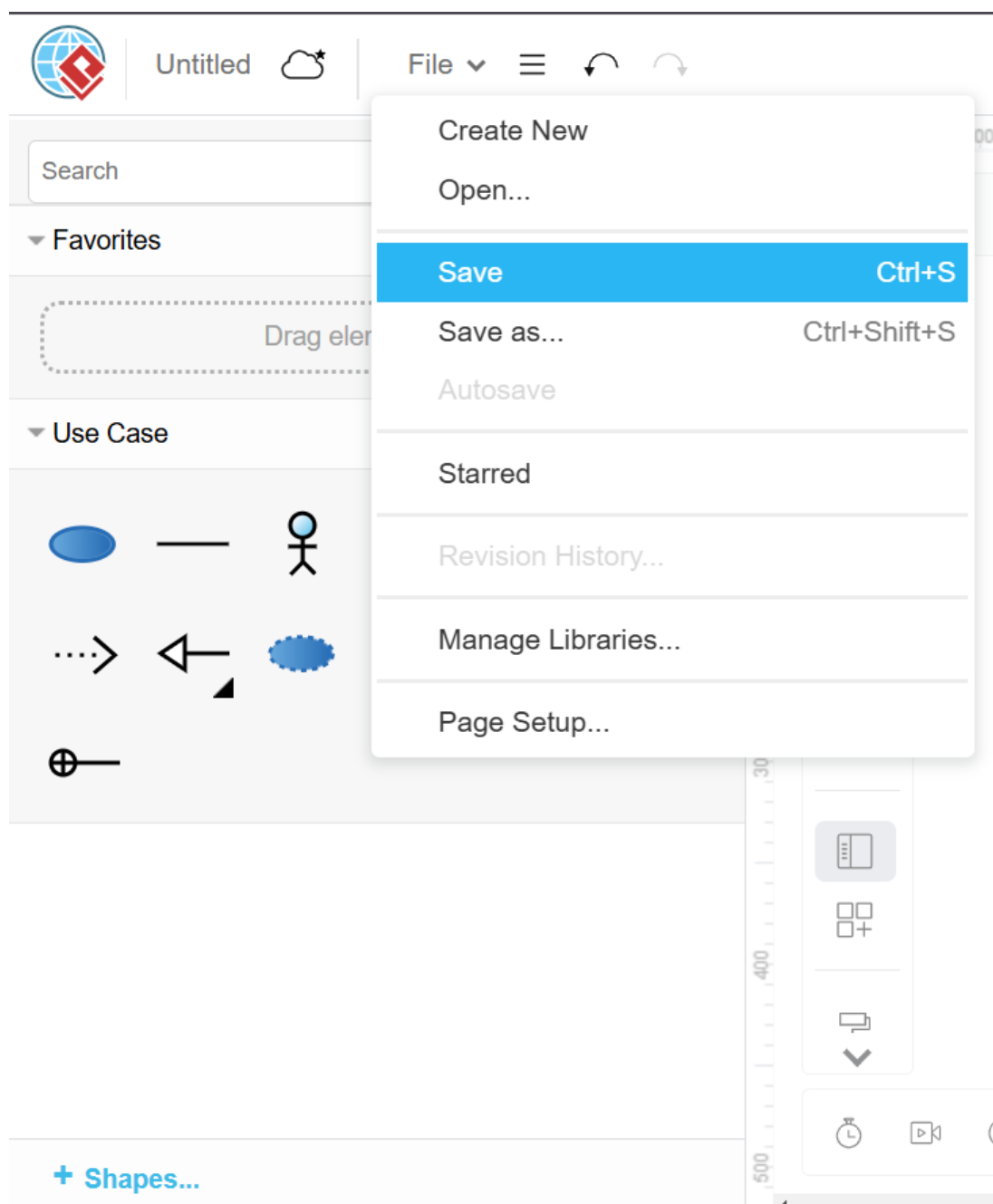
Przez kliknięcie na element prawym przyciskiem myszy uzyskujemy dostęp do opcji elementu takich jak skopiowanie czy usuwanie.



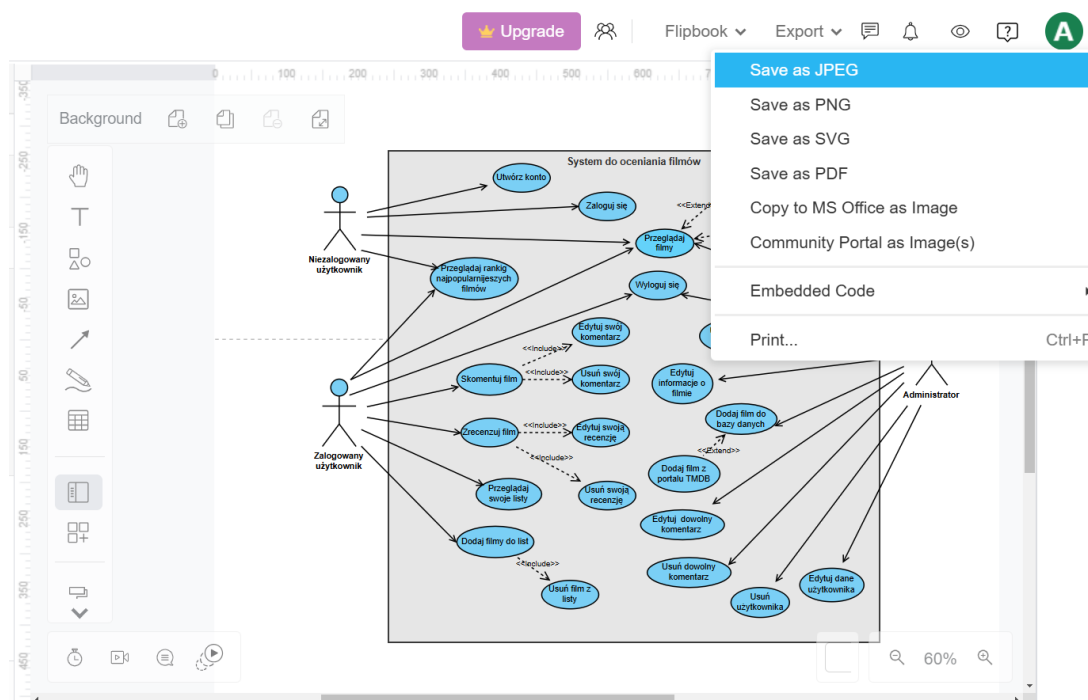
Aby zmienić tytuł projektu należy kliknąć na jego dotychczasowy tytuł który znajduje się w lewym górnym rogu strony.



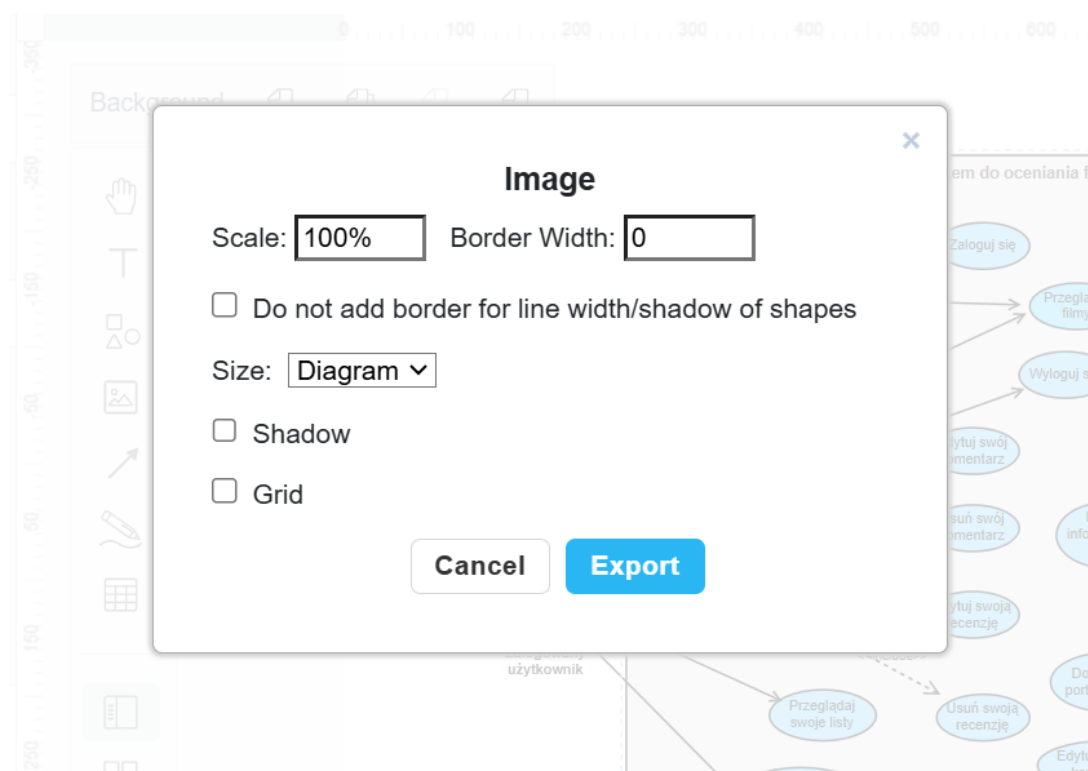
Aby zapisać projekt, tak aby później mieć do niego dostęp na swoim profilu, należy kliknąć na „File” w lewym górnym rogu i wybrać opcję „Save”.



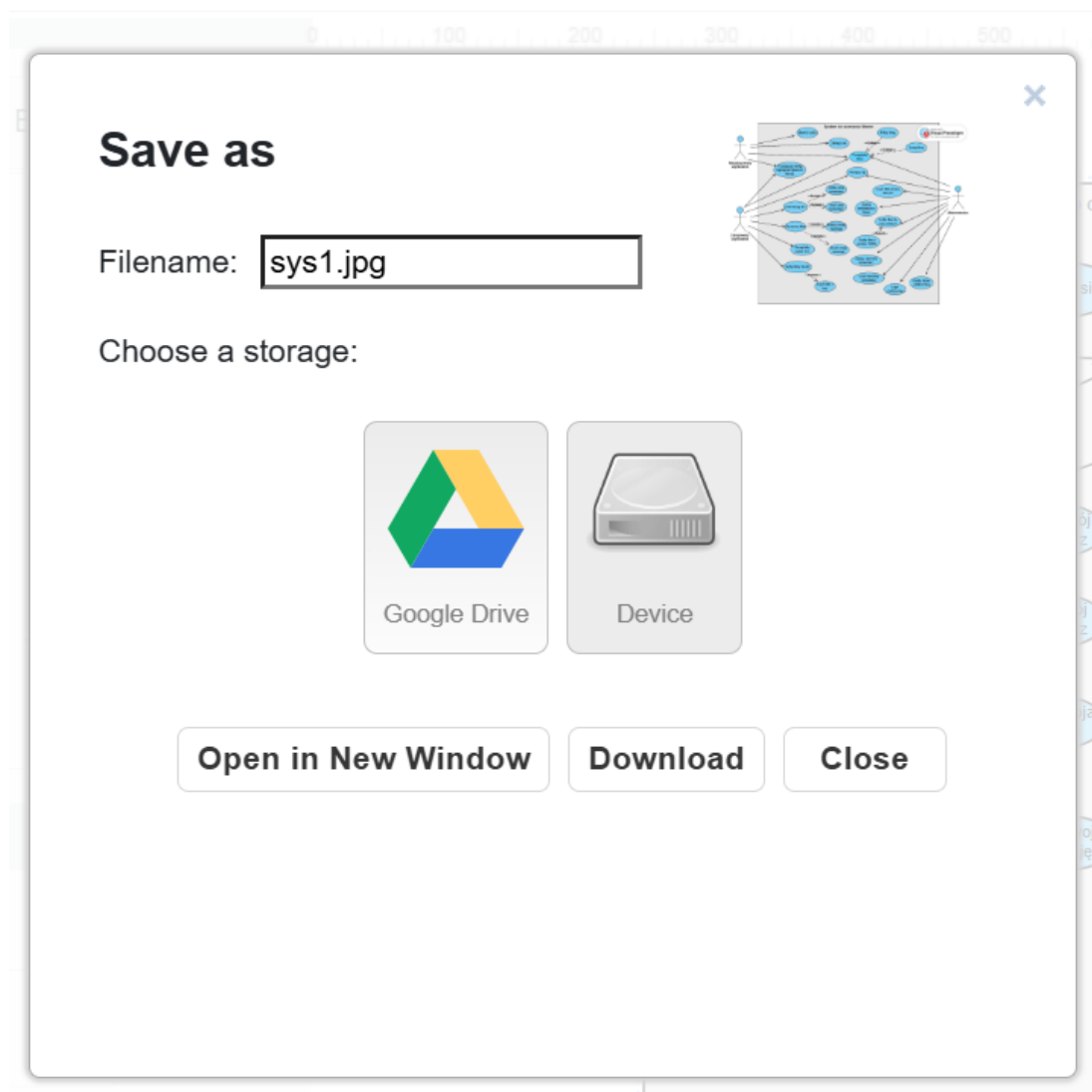
Aby wyeksportować plik w formie pliku graficznego należy kliknąć „Export” i wybrać pożądany format.



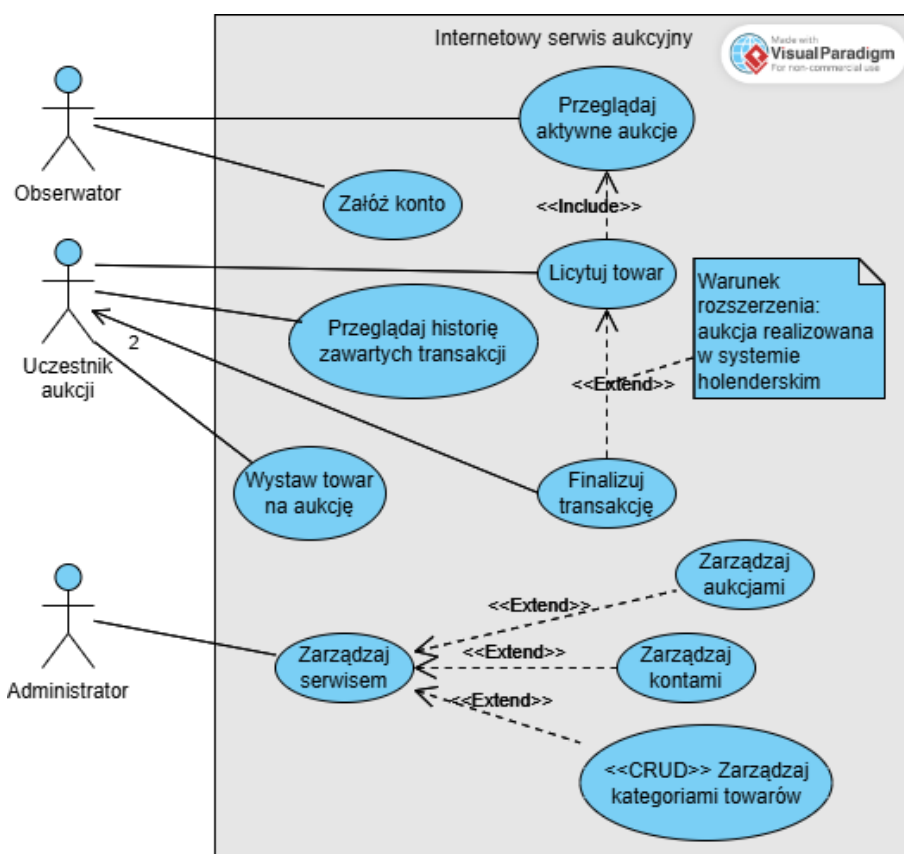
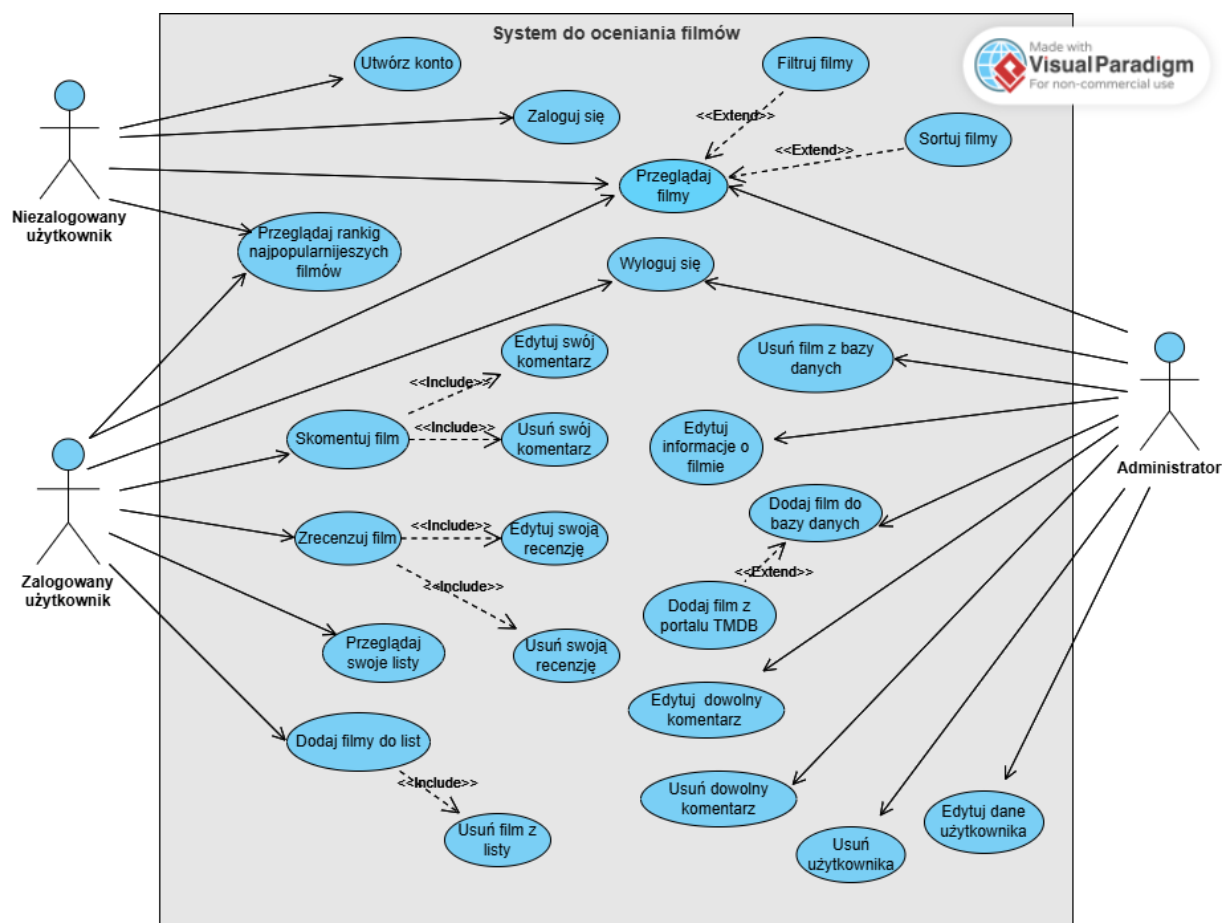
Można dostosować opcję pliku graficznego taki jak rozmiar czy grubość obramowania. Klikamy „Export”.



Następnie wybieramy miejsce, w którym chcemy zapisać plik. Tutaj wybrano zapis na lokalnym dysku („Device”).



Przy pomocy VisualParadigm stworzono poniższe diagramy:



9 Wnioski

Język UML jest potężnym i wszechstronnym narzędziem wspierającym projektowanie i dokumentowanie złożonych systemów informatycznych. Dzięki bogatej gamie diagramów, UML umożliwia modelowanie różnych aspektów systemu, od jego struktury po szczegółowe zachowanie i interakcje. Diagramy przypadków użycia pomagają w zrozumieniu wymagań, diagramy klas ułatwiają projektowanie struktury, a diagramy czynności i sekwencji pozwalają na szczegółową analizę przepływu danych i procesów. Użycie UML sprzyja komunikacji między zespołami, co jest kluczowe w projektach zespołowych, oraz zwiększa czytelność i utrzymanie dokumentacji technicznej. Dzięki standaryzacji UML jest uniwersalnym językiem zrozumiałym dla programistów, analityków i interesariuszy na całym świecie. Powstało wiele edytorów umożliwiających tworzenie diagramów. Jednym z nich jest Visual Paradigm online, który jest darmowy oraz wygodny w obsłudze.