

Instrukcja 3

Laboratoria 3, 4

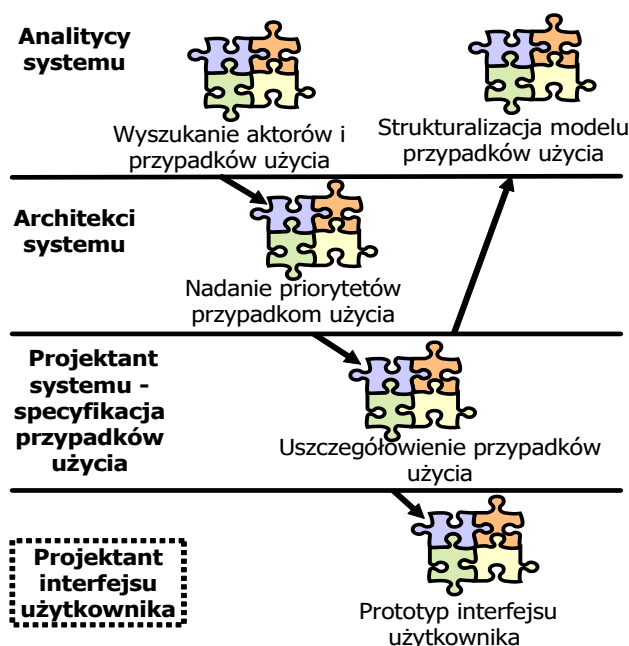
Specyfikacja wymagań funkcjonalnych za pomocą diagramu przypadków użycia

Cel laboratoriów:

Specyfikacja wymagań, zdefiniowanych w ramach laboratorium 2 (wg instrukcji 2), za pomocą diagramów przypadków użycia – tworzenie modelu przypadków użycia (informacja o diagramie: [wykład1](#), [wykład2](#), Dodatek 1, Dodatek 2, Dodatek 3).

Uwaga:

Za pomocą diagramów przypadków użycia należy modelować logikę biznesową procesów tzn. należy każdy scenariusz przypadków użycia traktować jako obsługę zdarzenia wywołania usługi, po wprowadzeniu danych. Poniżej, na rys. 1 przedstawiono poglądowo ten sposób podejścia podczas specyfikacji wymagań funkcjonalnych.



Rysunek. 1. Fragment cyklu życia oprogramowania

1. Należy kierować się zasadami podanymi na [wykładzie 2](#), dotyczącymi sposobu identyfikacji aktorów, określania granic tworzonego programowania, identyfikacji przypadków użycia, tworzenia scenariuszy przypadków użycia.
2. Grupa jednoosobowa powinna wykonać specyfikację wymagań za pomocą 1-2 złożonych przypadków użycia. Grupa dwuosobowa powinna zrealizować specyfikację wymagań za pomocą 2-3 złożonych przypadków użycia. Złożony przypadek użycia wykorzystuje powiązania z innymi przypadkami użycia (p. 6).
3. Należy stosować podsystemy w celu zaznaczenia odrębnych części systemu
4. Należy zdefiniować funkcje oprogramowania spełniające wymagania funkcjonalne tego oprogramowania, zdefiniowane w ramach laboratorium 2, uwzględniające ograniczenia wynikające z wymagań нефункциональных.

5. Uwaga szczegółowa: Należy podczas identyfikacji aktorów określić ich powiązania z przypadkami użycia stosując generalizację w przypadku powtórzeń w powiązaniach
 6. Uwaga szczegółowa: Podczas tworzenia scenariuszy poszczególnych przypadków użycia należy zwrócić uwagę na ewentualne powtarzające się w nich fragmenty, użyte obligatoryjnie lub opcjonalnie. W takich przypadkach należy umieścić te fragmenty w nowych przypadkach użycia i powiązać je z głównych przypadków użycia relacją:
 - 6.1. <<include>>, jeśli ten scenariusz jest wywoływany obligatoryjnie („zawsze musi wywołać”)
 - 6.2. <<extend>>, jeśli ten scenariusz jest wywoływany opcjonalnie („może, ale nie musi wywołać”) – ta relacja powinna być również użyta w przypadku jednorazowego wystąpienia takiego opcjonalnego podscenariusza
 - 6.3. <<use>>, jeśli ten scenariusz czasem musi być wywołany („czasem musi wywołać”) – ta relacja jest przykładem definicji relacji wykonanej przez programistę
 - 6.4. generalizacji w przypadku rozszerzania scenariusza.
 7. W ramach laboratorium 3 należy wykonać diagram przypadków użycia (DPU) na podstawie listy wymagań funkcjonalnych wykonanych podczas laboratorium 2 i rozpocząć wykonanie scenariuszy przypadków użycia.
 8. W ramach laboratorium 4 należy dokończyć scenariusze przypadków użycia. Można podczas wykonania scenariuszy dokonać refaktoryzacji DPU.
- Uwaga:** Należy rozwijać projekt wykonany przy realizacji instrukcji 2.

Przykłady w Dodatkach 1-3 instrukcji:

Dodatek 1- prezentacja listy wymagań funkcjonalnych i niefunkcjonalnych, diagramu przypadków użycia specyfikującym te wymagania funkcjonalne oraz scenariusze tych przypadków użycia sporządzone zgodnie z szablonem opisującym przypadki użycia (kontynuacja przykładu z instrukcji do lab2).

Dodatek 2 – prezentacja tworzenia diagramu przypadków użycia i wykorzystania formularzy **Info** i **Details** z pozycji **Open UseCase Details...** do umieszczenia opisu każdego z przypadków użycia, umieszczonych w Dodatku 1.

Dodatek3 – prezentacja wszystkich formularzy **Info** i **Details** dla pozostałych przypadków użycia przedstawionych w **Dodatek1**

Dodatek 1

1. Przykład specyfikacji wymagań - (z instrukcji 2, Dodatek 1):

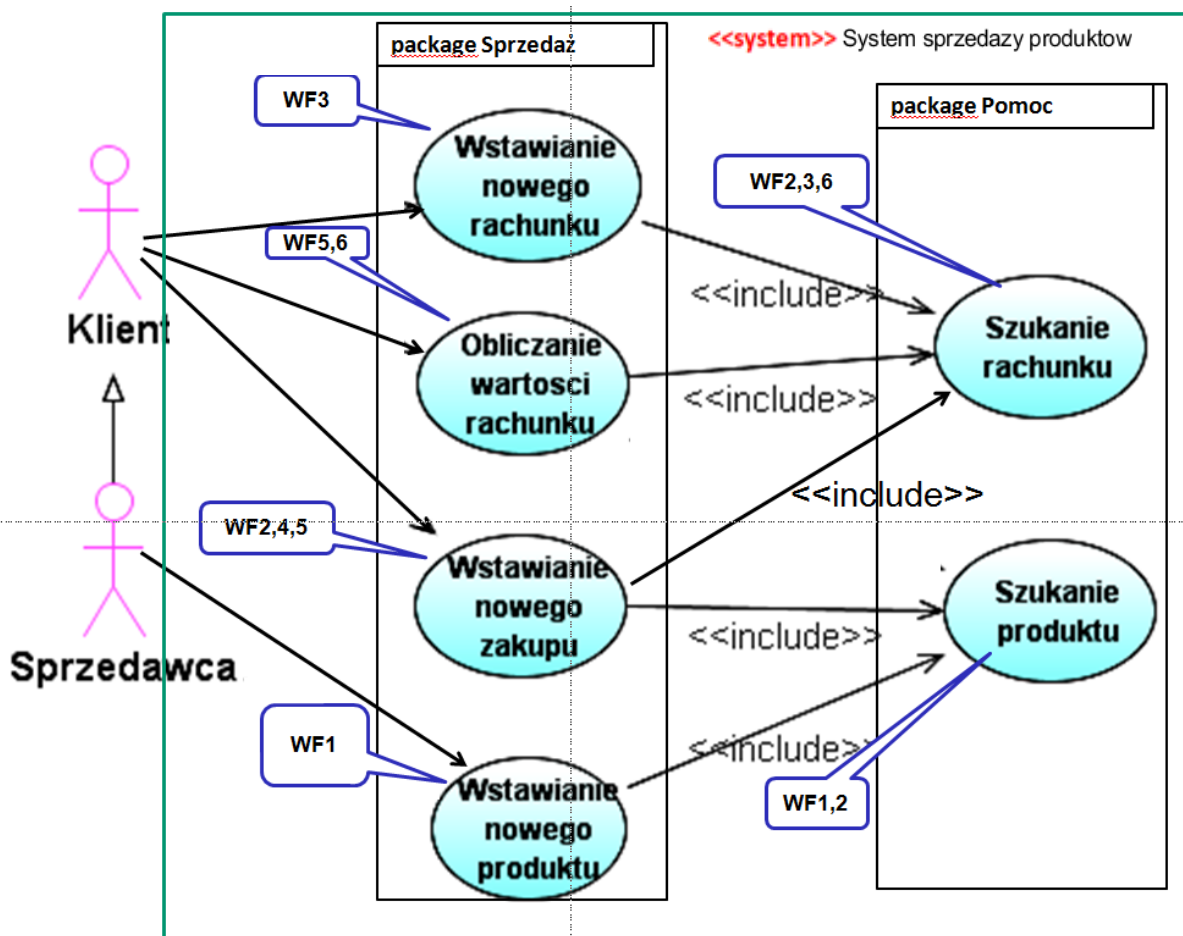
1.1. Lista wymagań funkcjonalnych tworzonego programu

1. System zawiera katalog produktów , do którego można wstawiać produkty
2. Można zakupić **cztery typy produktów** różniące się sposobem obliczania ceny detalicznej: bez promocji i bez podatku, z promocją i bez podatku, z podatkiem bez promocji, z podatkiem i z promocją,
3. Można wprowadzić wiele rachunków
4. Pozycje rachunku muszą zawierać produkty różne w sensie nazwy, ceny, podatku i promocji
5. Każda pozycja rachunku powinna podać swoją wartość brutto oraz dane produktu oraz ilość zakupionego produktu.
6. Na rachunku powinna znajdować się wartość łączna wszystkich zakupów oraz wartości zakupów należących do wybranych kategorii

1.2. Lista wymagań niefunkcyjnych programu programu

1. Wstawianie produktów może odbywać się tylko przez uprawnione osoby
2. Wstawianie nowych rachunków oraz wstawianie nowych zakupów jest dokonywane przez klientów
3. Zakupy mogą być dokonane przez Internet przez aplikację uruchamianą przez przeglądarkę lub bez jej pośrednictwa

2. Przykład specyfikacji przypadków użycia za pomocą diagramu przypadków użycia (DPU)



3. Przykład definicji aktorów

AKTOR	OPIS	PRZYPADKI UŻYCIA
Klient	<i>Klient może dokonywać zakupów wybranych produktów przez Internet korzystając z przeglądarki lub z aplikacji</i>	<ul style="list-style-type: none"> • PU Wstawianie nowego rachunku powiązane przez <<include>> z PU Szukanie rachunku • PU Obliczanie wartości rachunku powiązane przez <<include>> z PU Szukanie rachunku • PU Wstawianie nowego zakupu powiązane przez <<include>> z PU Szukanie rachunku oraz powiązane przez <<include>> z PU Szukanie produktu
Sprzedawca	<i>Sprzedawca może dodatkowo dodawać nowe produkty</i>	<ul style="list-style-type: none"> • PU Wstawianie nowego rachunku powiązane przez <<include>> z PU Szukanie rachunku • PU Obliczanie wartości rachunku powiązane przez <<include>> z PU Szukanie rachunku • PU Wstawianie nowego zakupu powiązane przez <<include>> z PU Szukanie rachunku oraz powiązane przez <<include>> z PU Szukanie produktu • PU Wstawianie nowego produktu powiązane przez <<include>> z PU Szukanie produktu

4. Przykłady opisu przypadków użycia (wraz z definicją scenariuszy) (PU – przypadek użycia)

PU Szukanie produktu

OPIS

CEL: Poszukiwanie produktu

WS (warunki wstępne): może być wywołany z **PU Wstawianie nowego produktu**, **PU Wstawianie nowego zakupu**

WK (warunki końcowe): podanie produktu o podanych atrybutach obowiązkowych: nazwa i cena oraz jeśli jest to wymagane: z podatkiem i promocją lub komunikat o braku produktu

PRZEBIEG:

1. Szukanie produktu przebiega według atrybutów: nazwy i ceny (obowiązkowo) oraz podatku i promocji, jeśli jest to wymagane zgodnie z danymi podanymi do przypadku użycia
2. Jeśli istnieje produkt o podanych atrybutach, zwracany jest produkt, w przeciwnym wypadku zwracana jest informacja o braku produktu.

PU Wstawianie nowego produktu

OPIS

CEL: Wstawienie nowego produktu

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): dodanie produktu o podanych atrybutach obowiązkowych: nazwa i cena oraz jeśli jest to wymagane: z podatkiem i promocją, jeśli nie było takiego produktu

PRZEBIEG:

1. Należy podać atrybuty produktu: nazwę, cenę jako obowiązkowe dane oraz podatek i promocję, jeśli jest to wymagane
2. Należy wywołać **PU Szukanie produktu**. Należy sprawdzić, czy produkt o podanych atrybutach już istnieje. Jeśli tak, należy zakończyć PU, w przeciwnym wypadku należy wstawić nowy produkt.

PU Szukanie rachunku

OPIS

CEL: Poszukiwanie rachunku

WS (warunki wstępne): może być wywołany z **PU Wstawianie nowego rachunku**, **PU Wstawianie nowego zakupu**, **PU Obliczanie wartości rachunku**

WK (warunki końcowe): podanie rachunku o podanym numerze lub komunikat o braku rachunku

PRZEBIEG:

1. Szukanie rachunku przebiega według numeru podanego do przypadku użycia
2. Jeśli istnieje rachunek o podanym numerze, zwracany jest rachunek, w przeciwnym wypadku zwracana jest informacja o braku rachunku.

PU Wstawianie nowego rachunku

OPIS

CEL: Wstawienie nowego rachunku

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): dodanie rachunku o podanym numerze, jeśli jest to unikatowy numer

PRZEBIEG:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku
2. Należy wywołać **PU Szukanie rachunku** w celu sprawdzenia, czy numer rachunku się powtarza.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, można wstawić nowy rachunek i zakończyć PU, w przeciwnym wypadku należy zakończyć PU bez wstawiania nowego rachunku.

PU Wstawianie nowego zakupu

OPIS

CEL: Wstawianie nowego zakupu

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): dodanie nowego zakupu o podanych atrybutach lub zwiększenie ilości zakupionego produktu, jeśli już taki produkt zakupiono lub komunikat o braku rachunku

PRZEBIEG:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku. Należy wybrać produkt oraz ilość zakupionego produktu.
2. Należy wywołać **PU Szukanie rachunku** w celu sprawdzenia, czy istnieje rachunek o podanym numerze.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, nie można wstawić nowego zakupu do rachunku i należy zakończyć PU, w przeciwnym wypadku należy wstawić nowy zakup
5. Należy wywołać **PU Szukanie produktu**. Jeśli wybrany produkt nie istnieje, należy zakończyć PU. **W przeciwnym przypadku należy wstawić nowy zakup do rachunku, przeglądając, czy istnieje już zakup z takim samym produktem. Jeśli istnieje, nie tworzy się nowego zakupu, tylko powiększa się ilość zakupu istniejącego o ilość nowego zakupu, w przeciwnym przypadku wstawia się nowy zakup.**

PU Obliczanie wartości rachunku

OPIS

CEL: Obliczanie wartości rachunku wg podanego podatku

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

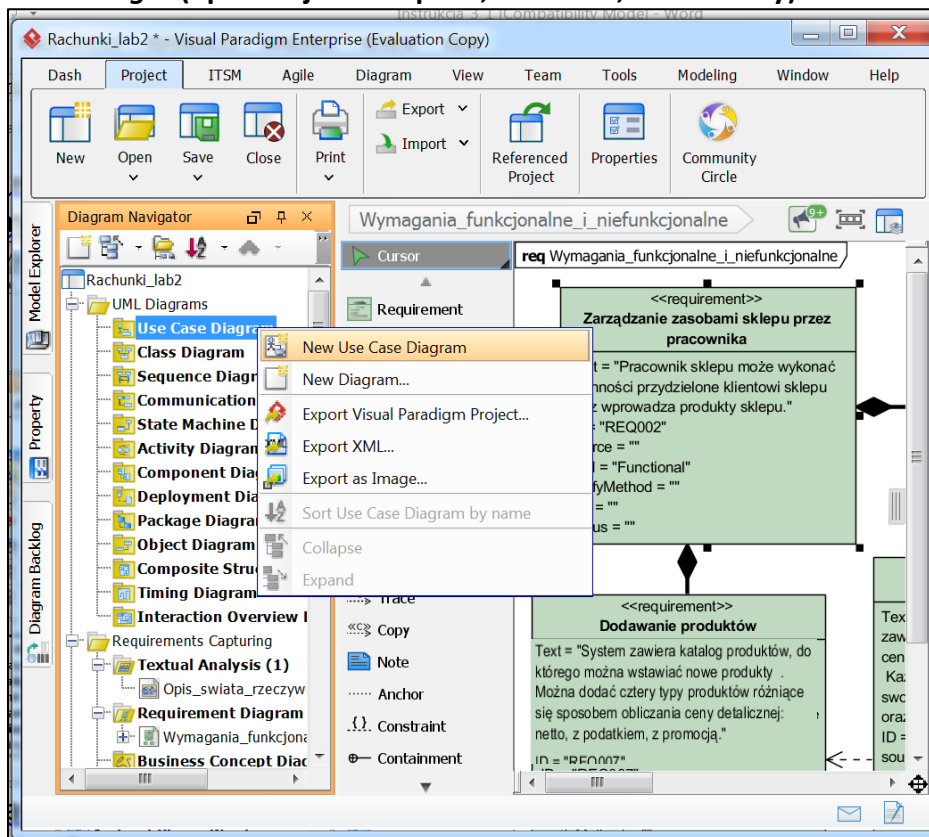
WK (warunki końcowe): podanie wartości całego rachunku o podanym numerze i parametrze wejściowym równym -2 lub wartości zakupionych towarów wg podanej kategorii podatku lub komunikat o braku rachunku

PRZEBIEG:

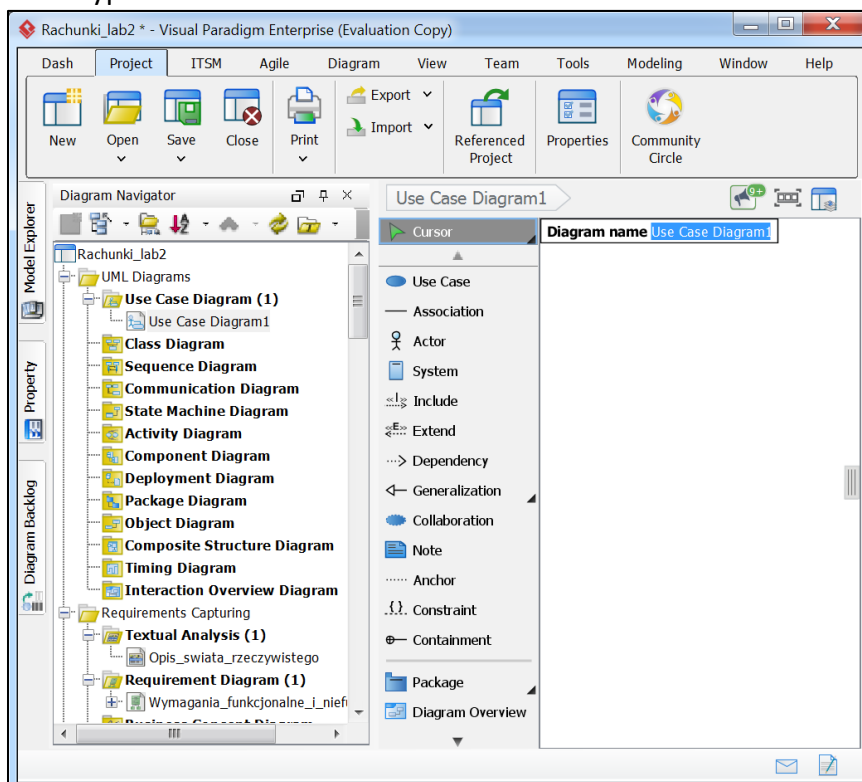
1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku oraz wartość podatku lub wartość -2
2. Należy wywołać **PU Szukanie rachunku** w celu sprawdzenia, czy rachunek o podanym numerze istnieje.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, nie można obliczyć wartości wybranego rachunku i należy zakończyć PU, w przeciwnym wypadku należy obliczyć wartość rachunku
4. Należy uruchomić pętlę, w której sumowane są wartości zakupu obliczane jako iloczyn ceny jednostkowej zakupionego produktu i ilości zakupu. Jeśli zachodzi potrzeba sumowania wartości zakupu zależna od wysokości podatku, należy podać wartość podatku i sumować jedynie zakupy o podanym podatku, w przeciwnym wypadku sumowane są wszystkie zakupy, gdy zamiast podatku zostanie przekazana wartość z poza zakresu wartości podatku np. równa -2.

Dodatek 2

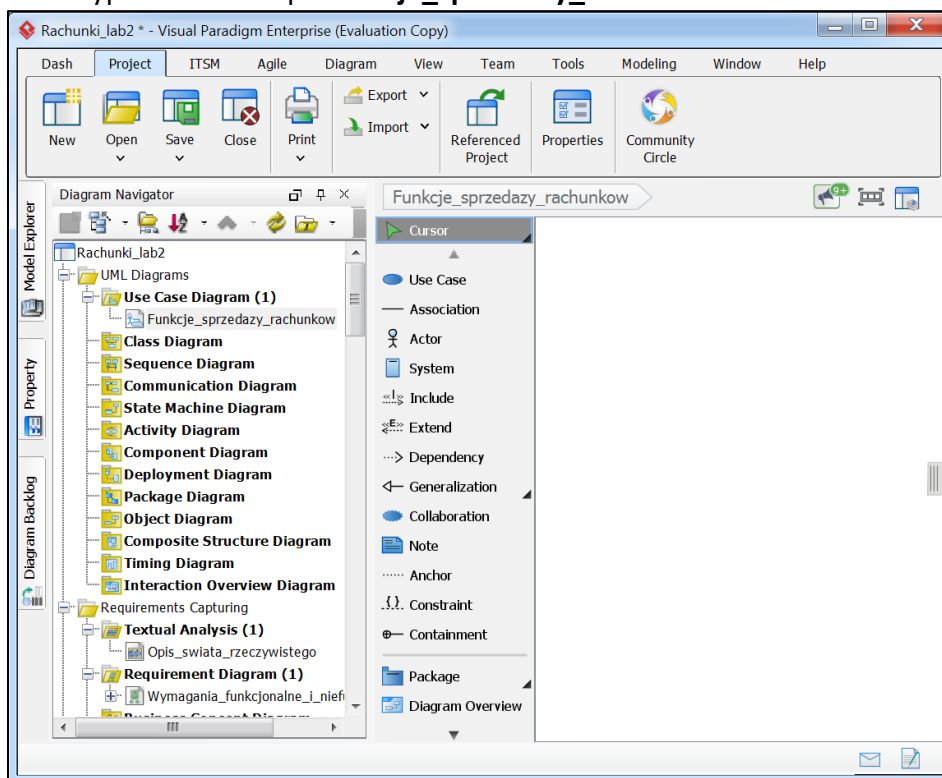
1. Tworzenie diagramów przypadków użycia (DPU) w wybranym środowisku np. *Visual Paradigm* (np. wersje: Enterprise, Standard, Community).



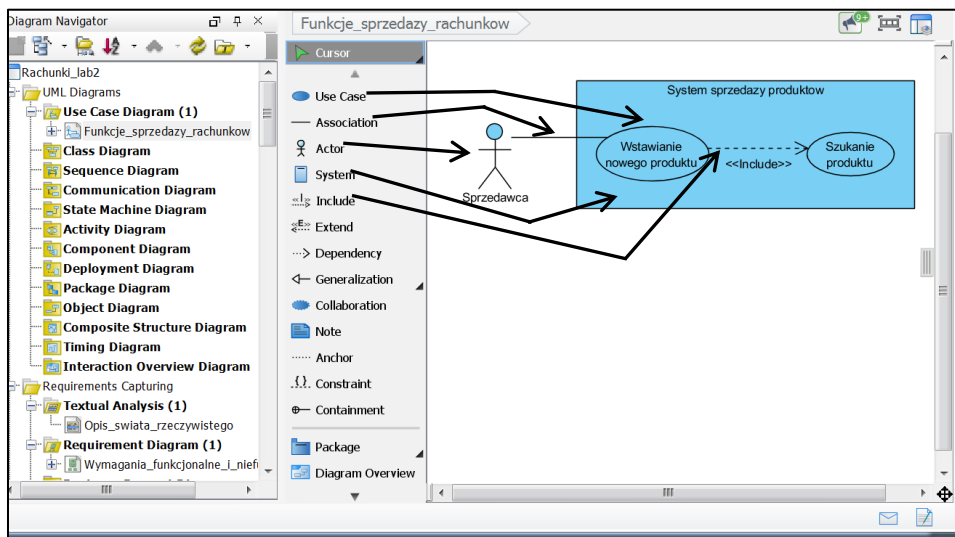
1.1. Wstawianie do projektu diagramu typu *Use Case* (DPU)– nadanie nazwy diagramowi typu *Use Case*



1.2. Wstawianie do projektu diagramu typu *Use Case* (DPU)– nadanie nazwy diagramowi typu *Use Case* np..**Funkcje_sprzedazy_rachunkow**

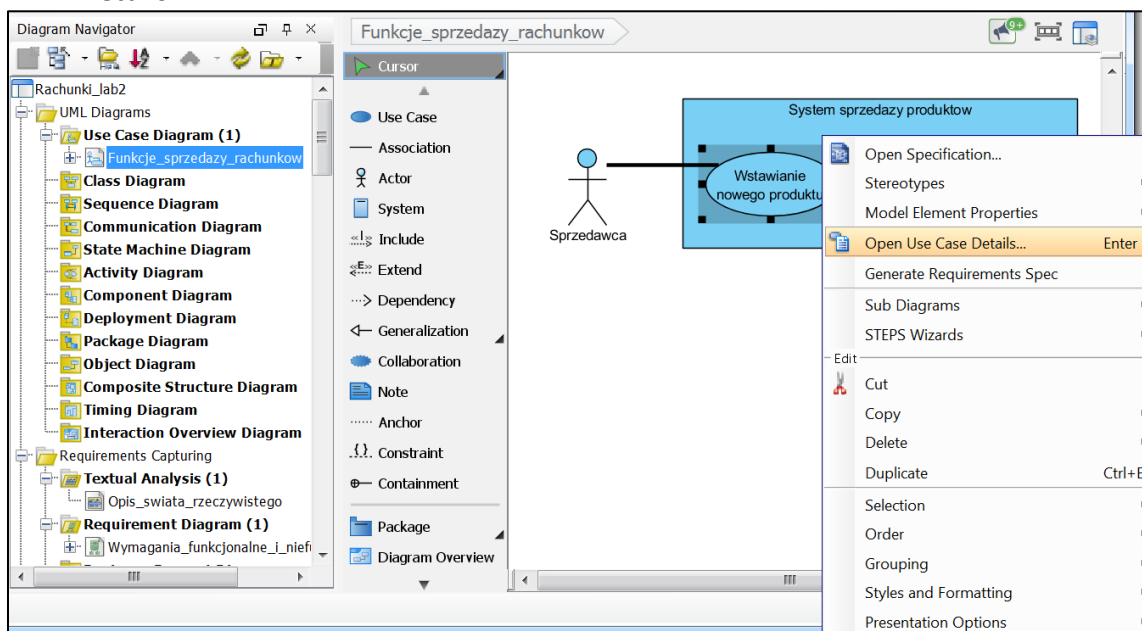


1.3. Wstawianie do projektu diagramu typu *Use Case* (DPU)– definicja przypadków użycia specyfikujących wymagania stawiane aplikacji w zakresie wstawiania produktów

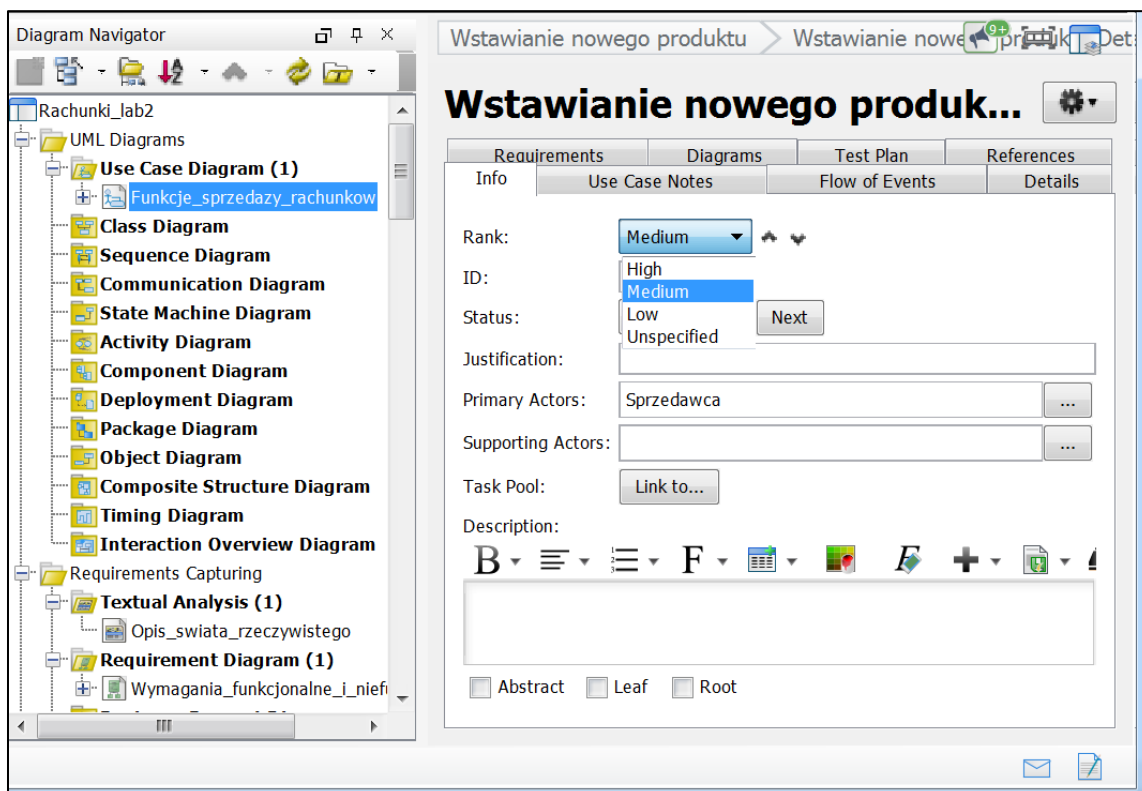


Przecignięcie ikon Actor, System, Use Case, pobranych z palety lewym klawiszem myszy i upuszczenie na diagramie przypadków użycia (DPU). Należy nadać im podane nazwy. Następnie, należy połączyć element Actor z przypadkiem użycia PU **Wstawianie nowego produktu** relacją Association, przeciągając ją z palety (z lewej strony) lewym klawiszem myszy i następnie położyć ją na elemencie Actor i przeciągnąć na PU **Wstawianie nowego produktu**. Podobnie należy połączyć przypadki użycia relacją <<Include>>, przeciągając ją z palety - i następnie należy ją położyć na PU **Wstawianie nowego produktu** i przeciągnąć do PU **Szukanie produktu**.

1.4. Definiowanie elementów typu *Use Case* – po kliknięciu prawym klawiszem myszy na PU **Wstawianie nowego produktu** należy dokonać wyboru z listy opcji *Open Use Case Details*



1.5. Nadanie wagi diagramowi – wybór wartości z listy w polu *Rank*



1.6. Wpisanie do podformularza *Info* w części *Description* scenariusza przypadku użycia Wstawianie nowego produktu

Diagram Navigator

Wstawianie nowego produktu > Wstawianie nowego produktu Details

Wstawianie nowego produktu

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Rank: Medium

ID: UC01

Status: Identify Next

Justification:

Primary Actors: Sprzedawca

Supporting Actors:

Task Pool: Link to...

Description:

PRZEBIEG:

1. Należy podać atrybuty produktu: nazwę, cenę jako obowiązkowe dane oraz podatek i promocje, jeśli jest to wymagane
2. 2. Należy wywołać PU Szukanie produktu. Należy sprawdzić, czy produkt o podanych atrybutach już istnieje. Jeśli tak, należy zakończyć PU, w przeciwnym wypadku należy wstawić nowy produkt.

Abstract Leaf Root

1.7. Wybór podformularza *Details* związanego z wybranym wcześniej PU

Diagram Navigator

Wstawianie nowego produktu > Wstawianie nowego produktu Details

Wstawianie nowego produktu

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Level:

Complexity:

Use Case Status:

Implementation Status:

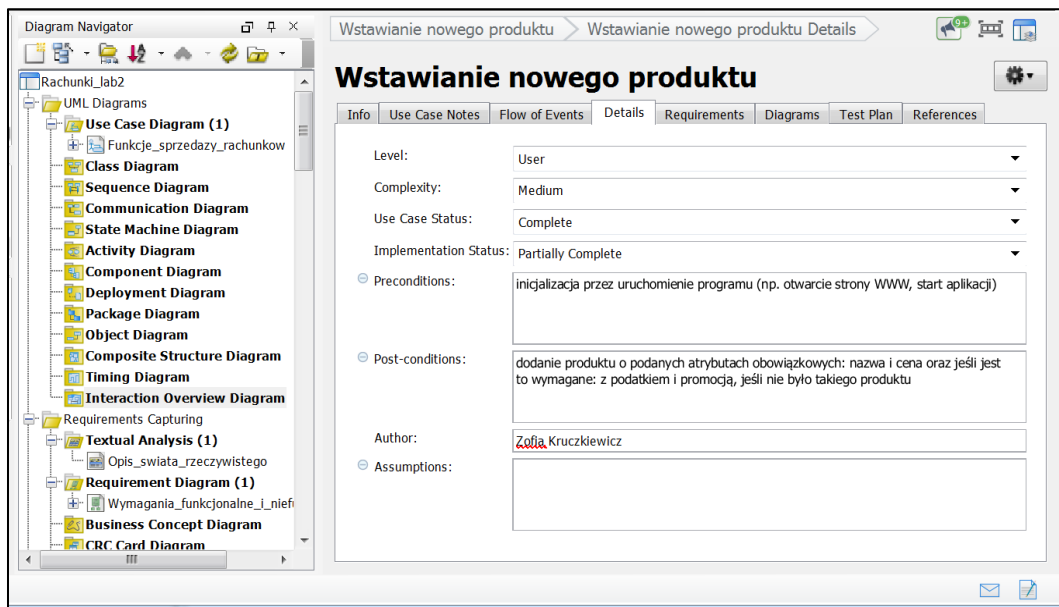
Preconditions:

Post-conditions:

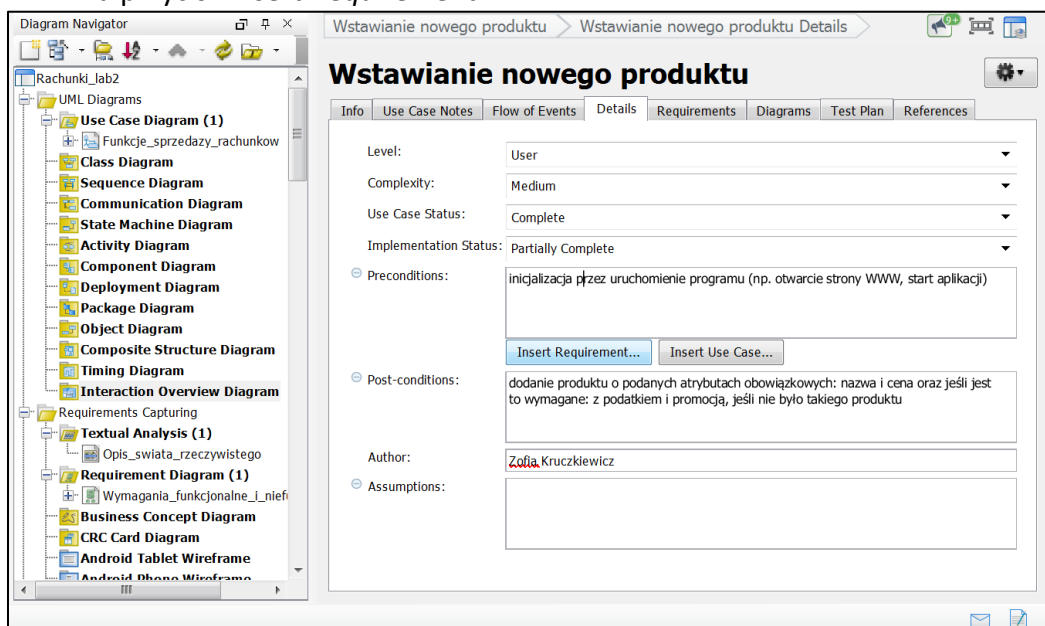
Author:

Assumptions:

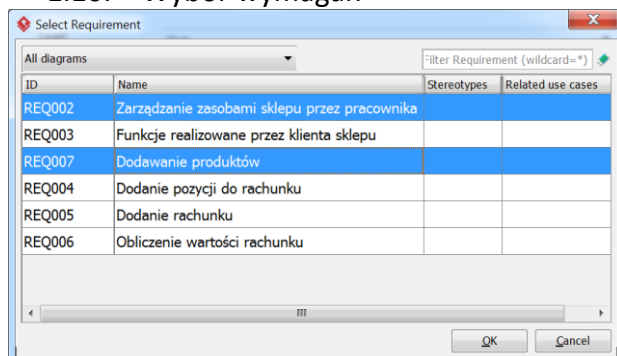
1.8. Wybór podformularza *Details* związanego z wybranym wcześniej PU – nadanie wartości poszczególnym polom formularza przez wybór z listy lub wprowadzenie tekstu

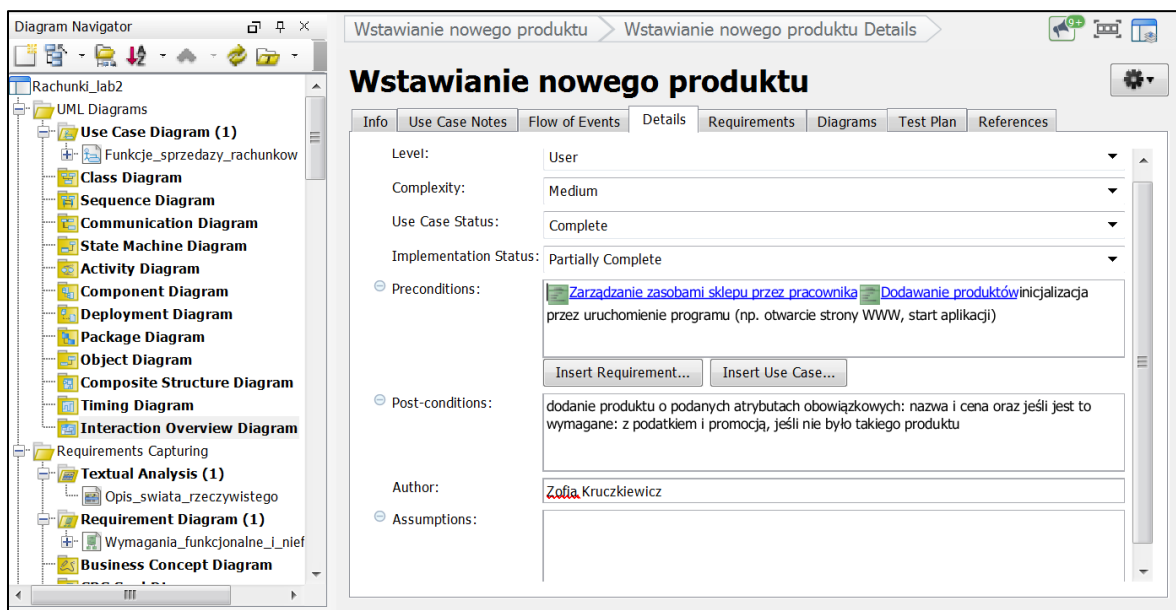


1.9. Powiązanie *Precondition* z wybranymi wymaganiami z diagramu wymagań (*Requirements Diagram*) – należy kliknąć na powierzchnię pola Precondition i kliknąć na przycisk *Insert Requirement...*

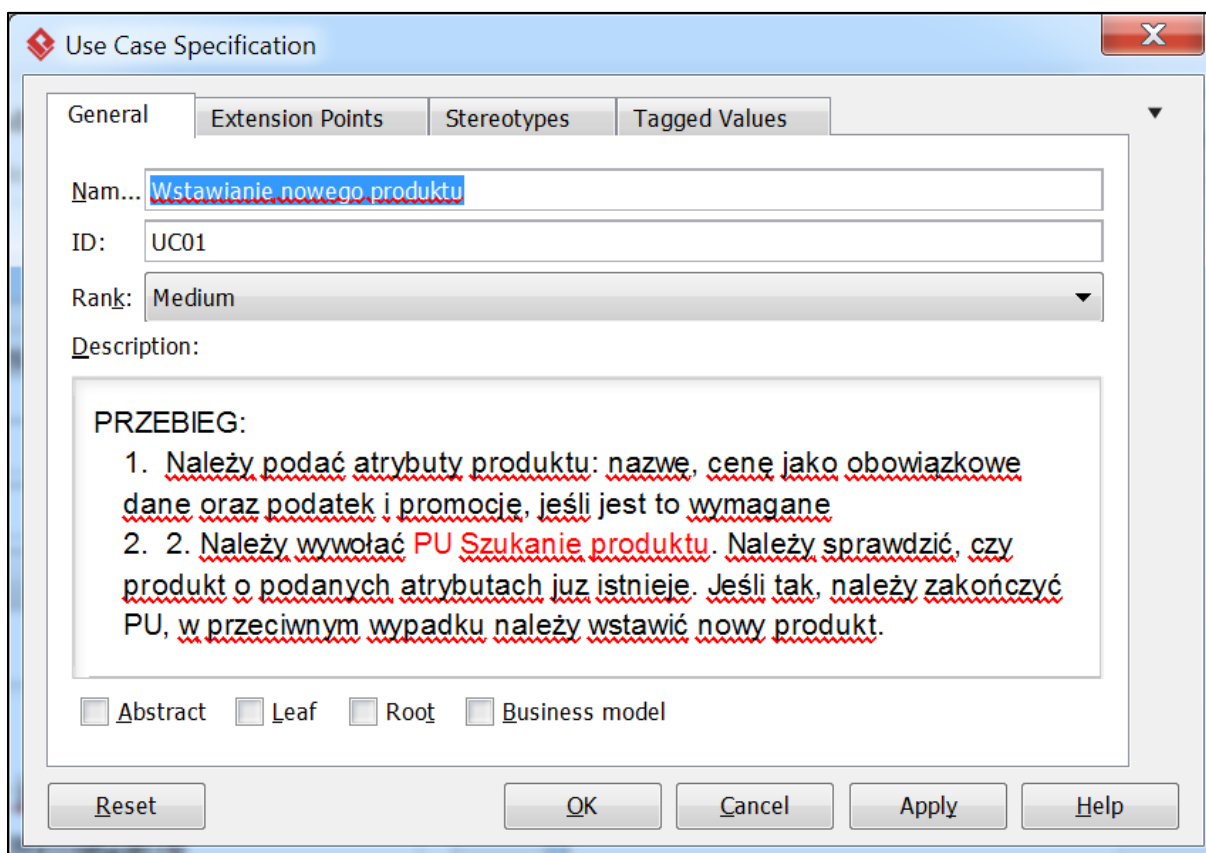


1.10. Wybór wymagań

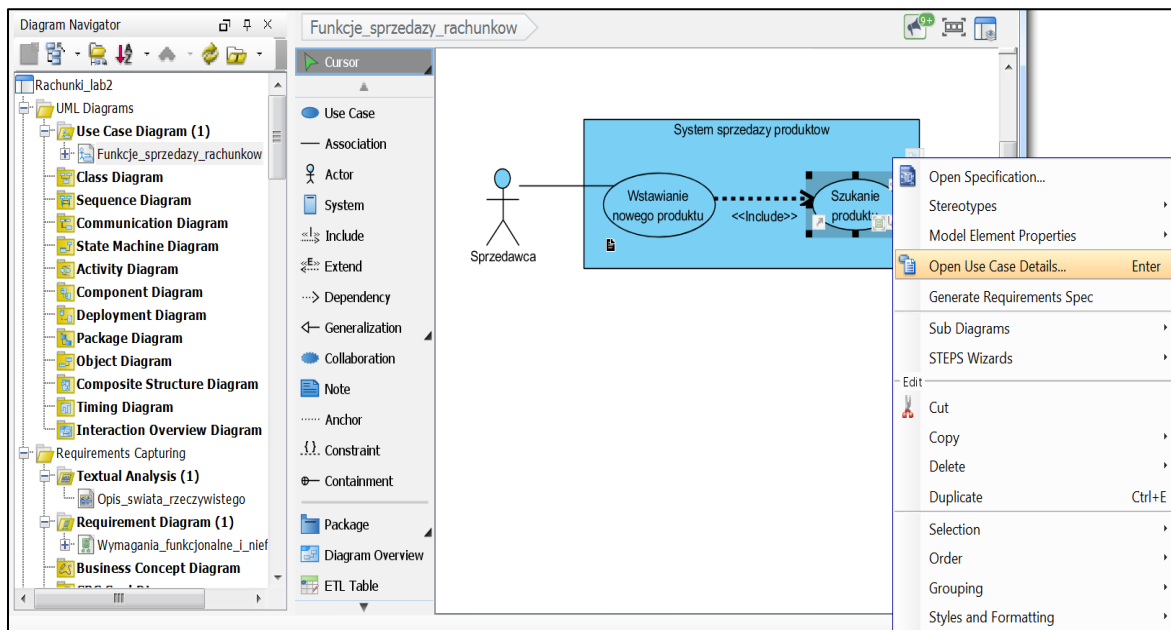


1.11. Rezultat dodania wybranego wymagania do pola *Precondition*

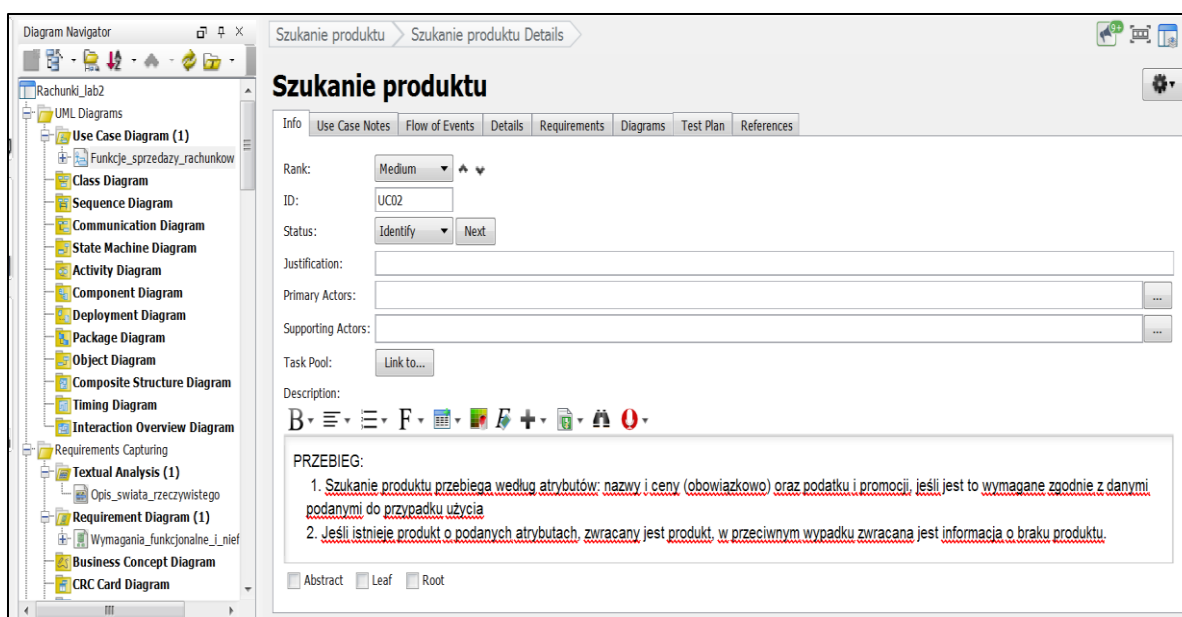
1.12. Po kliknięciu prawym klawiszem myszy na PU **Wstawianie nowego produktu** należy wybrać z listy opcję *Open Specification* – powinien ukazać tekst wprowadzony wcześniej w podformularzu *Info* opcji *Open Use Case Details*



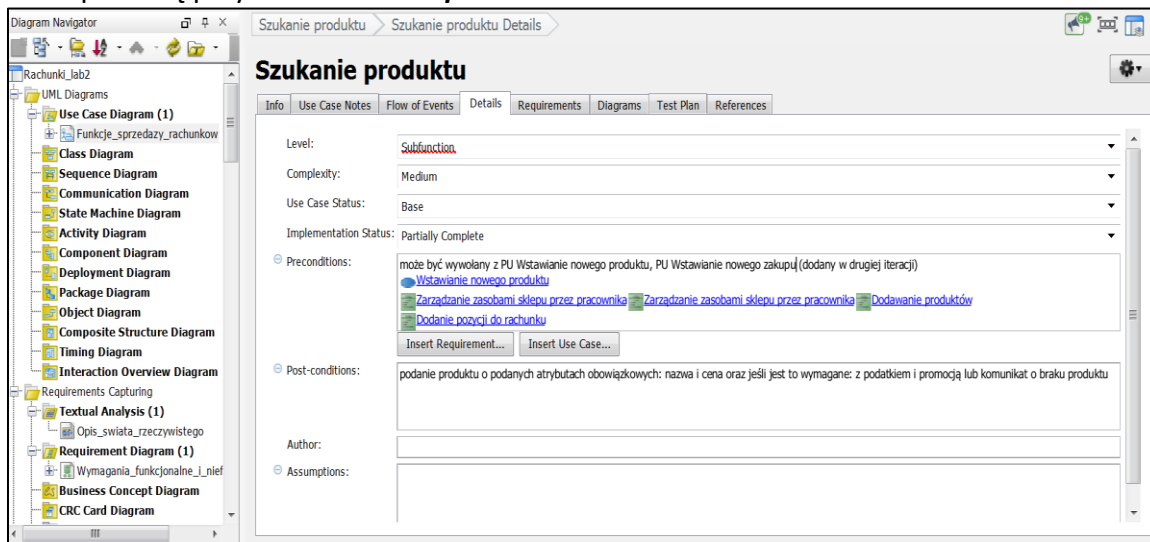
1.13. Definiowanie elementów typu Use Case – po kliknięciu prawym klawiszem myszy na PU Szukanie produktu należy dokonać wyboru z listy opcji *Open Use Case Details*



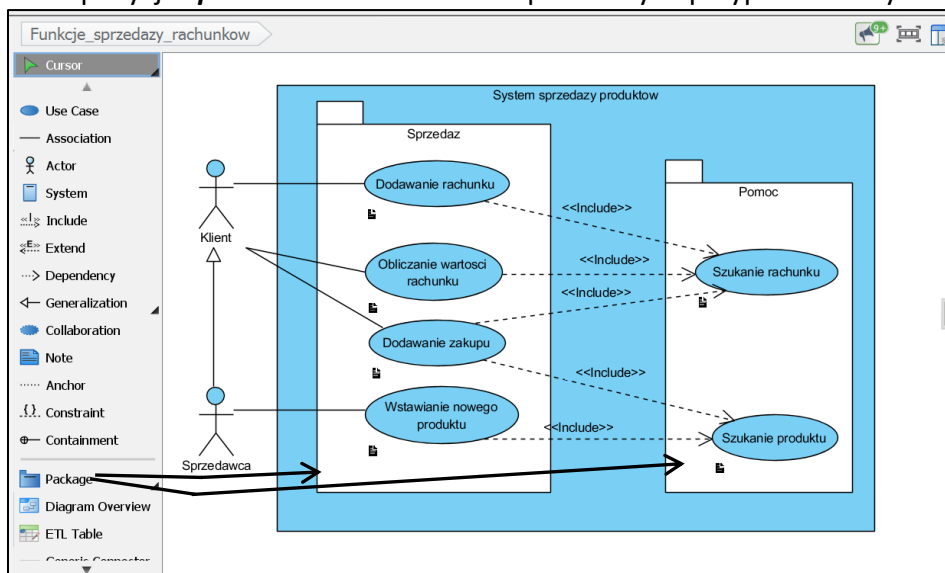
1.14. Wpisanie do podformularza *Info* (zakładka Info) w części *Description* scenariusza przypadku użycia PU Szukanie produktu



- 1.15. Wybór podformularza *Details* związanego z wybranym wcześniej PU – nadanie wartości poszczególnym polom formularza przez wybór z listy lub wprowadzenie tekstu. Dodanie przypadków użycia, z których wywoływany jest **PU Szukanie produktu** - kliknięciu na przycisk **Insert Use Case...** i wyborze przypadku użycia z listy, z którego wywoływany jest **PU Szukanie produktu**. Dodanie warunków początkowych i końcowych oraz powiązanych (i specyfikowanych) wymagań do pola *Precondition* za pomocą przycisku **Insert Requirement...**



- 1.16. Końcowa wersja diagramu przypadków użycia – elementy **Sprzedaz** oraz **Pomoc** typu **package** grupują przypadki użycia główne oraz pomocnicze. W **Dodatek 3** umieszczone są podformularze **Info** i **Details** dostępne z formularza dostępnego z pozycji **Open Use Case Details** dla pozostałych przypadków użycia.



2. Dodatkowa pomoc ze strony Visual Paradigm:

- 2.1. Pomoc: [Drawing use case diagrams.](http://www.visual-paradigm.com/support/documents/vpumluserguide/94/2575/6362_drawinguseca.html)
- 2.2. Pomoc: [Use case diagram notations guide](http://www.visual-paradigm.com/support/documents/vpuserguide/94/2575/84257_usecasediagr.html)
- 2.3. Pomoc: [Documenting use case details.](http://www.visual-paradigm.com/support/documents/vpumluserguide/94/2575/21179_documentingu.html)
- 2.4. Pomoc: [Creating requirement diagram.](http://www.visual-paradigm.com/support/documents/vpuserguide/94/158/6516_creatingrequ.html)

Dodatek3

Podformularze **Info** i **Details** dostępne z formularza dostępnego z pozycji **Open Use Case Details** dla pozostałych przypadków użycia.

1. PU Dodawanie rachunku

Dodawanie rachunku > Dodawanie rachunku Details

Dodawanie rachunku

Info
Use Case Notes
Flow of Events
Details
Requirements
Diagrams
Test Plan
References

Rank: ▲ ▼

ID:

Status:

Justification:

Primary Actors: ...

Supporting Actors: ...

Task Pool:

Description:

PRZEBIEG:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku
2. Należy wywołać PU Szukanie rachunku w celu sprawdzenia, czy numer rachunku się powtarza.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, można wstawić nowy rachunek i zakończyć PU, w przeciwnym wypadku należy zakończyć PU bez wstawiania nowego rachunku.

Abstract Leaf Root

Dodawanie rachunku > Dodawanie rachunku Details

Dodawanie rachunku

Info
Use Case Notes
Flow of Events
Details
Requirements
Diagrams
Test Plan
References

Level:

Complexity:

Use Case Status:

Implementation Status:

Preconditions: [Zarządzanie zasobami sklepu przez pracownika](#)
 [Funkcje realizowane przez klienta sklepu](#) [Dodanie rachunku](#)

Post-conditions:

Author:

Assumptions:

2. PU Dodawanie zakupu

Dodawanie zakupu > Dodawanie zakupu Details

Dodawanie zakupu

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Rank: Medium

ID: UC05

Status: Identify Next

Justification:

Primary Actors: Klient

Supporting Actors:

Task Pool: Link to...

Description:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku. Należy wybrać produkt oraz ilość zakupionego produktu.

2. Należy wywołać PU Szukanie rachunku w celu sprawdzenia, czy istnieje rachunek o podanym numerze.

3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, nie można wstawić nowego zakupu do rachunku i należy zakończyć PU, w przeciwnym wypadku należy wstawić nowy zakup.

5. Należy wywołać PU Szukanie produktu. Jeśli wybrany produkt nie istnieje, należy zakończyć PU. W przeciwnym przypadku należy wstawić nowy zakup do rachunku, przeglądając, czy istnieje już zakup z takim samym produktem. Jeśli istnieje, nie tworzy się nowego zakupu, tylko powiększa się ilość zakupu istniejącego o ilość nowego zakupu, w przeciwnym przypadku wstawia się nowy zakup.

Abstract Leaf Root

Dodawanie zakupu > Dodawanie zakupu Details

Dodawanie zakupu

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Level: User

Complexity: Medium

Use Case Status: Base

Implementation Status: Started

Preconditions:

- inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)
- Zarządzanie zasobami sklepu przez pracownika
- Funkcje realizowane przez klienta sklepu
- Dodanie pozycji do rachunku

Post-conditions:

- Dodanie nowego zakupu o podanych atrybutach lub zwiększenie ilości zakupionego produktu, jeśli już taki produkt zakupiono lub komunikat o braku rachunku

Author:

Assumptions:

3. PU Obliczanie wartosci rachunku

Obliczanie wartosci rachunku > Obliczanie wartosci rachunku Details

Obliczanie wartosci rachunku

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Rank: Medium

ID: UC04

Status: Identify Next

Justification:

Primary Actors: Klient

Supporting Actors:

Task Pool: Link to...

Description:

B

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku oraz wartość podatku lub wartość -2
2. Należy wywołać PU Szukanie rachunku w celu sprawdzenia, czy rachunek o podanym numerze istnieje
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, nie można obliczyć wartości wybranego rachunku i należy zakończyć PU, w przeciwnym wypadku należy obliczyć wartość rachunku
4. Należy uruchomić pętle, w której sumowane są wartości zakupu obliczane jako iloczyn ceny jednostkowej zakupionego produktu i ilości zakupu. Jeśli zachodzi potrzeba sumowania wartości zakupu zależna od wysokości podatku, należy podać wartość podatku i sumować jedynie zakupy o podanym podatku, w przeciwnym wypadku sumowane są wszystkie zakupy, gdy zamiast podatku zostanie przekazana wartość z poza zakresu wartości podatku np. równa -2.

Abstract Leaf Root

Obliczanie wartosci rachunku > Obliczanie wartosci rachunku Details

Obliczanie wartosci rachunku

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Level: User

Complexity: Medium

Use Case Status: Base

Implementation Status: Started

Preconditions:

- inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)
- Zarządzanie zasobami sklepu przez pracownika
- Funkcje realizowane przez klienta sklepu
- Obliczenie wartości rachunku

Post-conditions:

podanie wartości całego rachunku o podanym numerze i parametrze wejściowym równym -2 lub wartości zakupionych towarów wg podanej kategorii podatku lub komunikat o braku rachunku

Author:

Assumptions:

4. PU Szukanie rachunku

Szukanie rachunku > Szukanie rachunku Details

Szukanie rachunku

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Rank: Medium

ID: UC06

Status: Identify Next

Justification:

Primary Actors: ...

Supporting Actors: ...

Task Pool: Link to...

Description:

PRZEBIEG:

1. Szukanie rachunku przebiega według numeru podanego do przypadku użycia
2. Jeśli istnieje rachunek o podanym numerze, zwracany jest rachunek, w przeciwnym wypadku zwracana jest informacja o braku rachunku.

Abstract Leaf Root

Szukanie rachunku > Szukanie rachunku Details

Szukanie rachunku

Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Level: Subfunction

Complexity: Medium

Use Case Status: Base

Implementation Status: Started

Preconditions: może być wywołany z PU Wstawianie nowego rachunku, PU Wstawianie nowego zakupu, PU Obliczanie wartości rachunku
[Dodawanie rachunku](#) [Dodawanie zakupu](#) [Obliczanie wartości rachunku](#)
[Dodanie pozycji do rachunku](#) [Dodanie rachunku](#) [Obliczenie wartości rachunku](#)

Post-conditions: podanie rachunku o podanym numerze lub komunikat o braku rachunku

Author:

Assumptions: