

Instrukcja 3

Laboratorium 3

**Wykonanie opisu biznesowego „świata rzeczywistego”
projektowanego oprogramowania, definicja wymagań
funkcjonalnych i нефункциональных projektowanego
oprogramowania**

Cel laboratorium:

Opis najważniejszych procesów biznesowych występujących w tzw. „świecie rzeczywistym”, które należy zautomatyzować za pomocą projektowanej aplikacji oraz specyfikacja wymagań funkcjonalnych i нефункциональных tej aplikacji (wykład3 , Dodatek 1 instrukcji)

1. Wybór tematu aplikacji:

- 1.1. Program obsługujący konto bankowe
- 1.2. Program realizujący edytor tekstu
- 1.3. Program obsługujący system alarmowy
- 1.4. Program obsługujący Dział Ewidencji Ludności
- 1.5. Program obsługujący system informacyjny linii autobusowych
- 1.6. Program obsługujący wypożyczalnię kaset video
- 1.7. Program obsługujący wypożyczalnię sprzętu turystycznego
- 1.8. Program obsługujący zakład transportowy
- 1.9. Program obsługujący zapisy na zajęcia na wyższych uczelniach
- 1.10. Program wystawiający rachunki
- 1.11. Program obsługujący zakład naprawy sprzętu gospodarstwa domowego
- 1.12. Program obsługujący biuro informacji turystycznej
- 1.13. Program obsługujący system rezerwacji miejsc w hotelach
- 1.14. Program obsługujący pasażerskie linie lotnicze

Uwagi:

- Użyto celowo słowo **program** określające temat projektu, ponieważ diagramy UML, stosowane podczas projektowania mają służyć do projektowania kodu programu napisanego w języku obiektowym.
- Można zaproponować **własny temat aplikacji** do realizacji.

2. Wykonanie opisu „świata rzeczywistego”, przekazywany przez stronę zamawiającą oprogramowanie, wg następującego schematu:**2.1. Opis zasobów ludzkich**

Co robią pracownicy? - Na podstawie tego opisu określa się po przystąpieniu do modelowania oprogramowania, jakie funkcje powinno wykonywać tworzone oprogramowanie, aby automatycznie wspomagać pracę ludzi,

Uwaga: w przypadku systemów wbudowanych opis „świata rzeczywistego” powinien dotyczyć procesów technologicznych, które należy zautomatyzować.

2.2. Przepisy i strategia firmy

Co ogranicza działalność firmy? – Wykonanie opisu zawierającego specyfikację przepisów, ustaw, zarządzeń, strategii firmy itp. Po przystąpieniu do modelowania oprogramowania należy uwzględnić te ograniczenia na działanie funkcji, jakie system powinien wykonywać, aby wspomagać automatycznie pracę ludzi,

2.3. Dane techniczne - opis warstwy technicznego systemu informacyjnego (rozmiary, architektura, liczba użytkowników, pozostałe funkcje programu, stosowane dotąd oprogramowanie i sprzęt):

- 2.3.1. Dane ilościowe związane z przetwarzaniem realizowanym przez projektowany program:
 - ilu pracowników,
 - ile danych,
 - jak często wykonuje się poszczególne operacje na danych,
- 2.3.2. Dane o lokalizacji firmy
- 2.3.3. Dane o profilach klientów firmy oraz pracowników firmy,
- 2.3.4. Dane o używanym sprzęcie i oprogramowaniu.

3. Zdefiniowanie wymagań funkcjonalnych i нефункциональных programu na podstawie opisu biznesowego, przedstawiające czynności projektowanego oprogramowania

- co program powinien robić, aby zautomatyzować procesy wykonywane przez tzw. zasoby ludzkie? – reprezentowane przez wymagania funkcjonalne
- jakie rozwiązania technologiczne zastosować, aby program spełniał oczekiwania strony zamawiającej w zakresie bezpieczeństwa (security), niezawodności (safety), skalowalności i wydajności.

4. Wykonany tekst należy umieścić w projekcie zawierającym model UML, jako dodatkowe elementy (np. diagram wymagań, mimo, że nie jest diagramem UML, opisy tekstowe itp) zgodnie z przykładem zamieszczonym w dodatku do instrukcji (Dodatek 2).

Dodatek 1

1. System sporządzania rachunków - opis biznesowy „świata rzeczywistego” przygotowany przez stronę zamawiającą.

1.1. Opis zasobów ludzkich

Pracownik firmy handlowej może dodawać do katalogu produktów nowe produkty. Każdy produkt jest reprezentowany przez następujące dane: nazwa, cena netto lub nazwa, cena netto, podatek od ceny netto lub nazwa, cena netto i promocja od ceny netto lub nazwa, cena netto, podatek od ceny netto i promocja od ceny netto. Klient może założyć rachunek nadając mu unikalny numer oraz może dodawać nowe zakupy. Każda pozycja zakupu powinna zawierać inny produkt. W przypadku dokonywania zakupu tego samego produktu, który znajduje się w innej pozycji zakupu, należy jedynie zwiększyć ilość tego produktu w tym istniejącym zakupie. Klient może sprawdzić, jaką wartość ma rachunek oraz wartości zakupionych produktów z poszczególnych grup podatkowych. Pracownik firmy handlowej może dodawać nowe typy produktów. Pracownik ten może dodatkowo modyfikować i usuwać dane o produktach i promocjach.

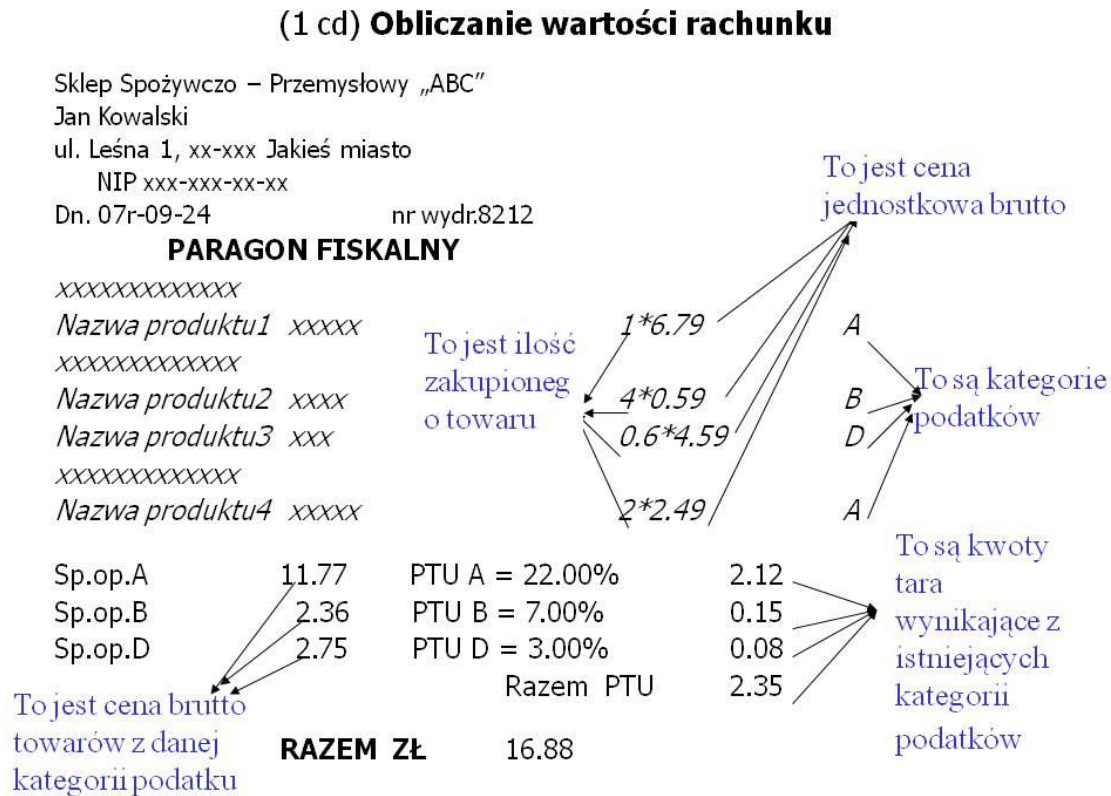
1.2. Przepisy, strategia firmy

Pracownik ponosi odpowiedzialność za poprawność danych - odpowiada materialnie za niezgodność danych ze stanem produktów firmy handlowej. System sporządzania rachunków powinien być przyjazny dla klienta biznesowego

1.3. Dane techniczne

Klient może przeglądać katalog sprzedaży za pośrednictwem strony internetowej lub bezpośrednio za pomocą specjalnego programu (tak obsługuje już wdrożone programy). Zakłada się, że klientów, jednocześnie przeglądających dane katalogu sprzedaży oraz dokonujących zakupu produktów może być ponad 1000 oraz firma handlowa może oferować kilkadziesiąt tysięcy typów produktów. Firma handlowa składa się z kilku ośrodków w różnych miastach na terenie kraju (lista miast jest dołączona do umowy). Zaleca się stosowanie technologii Java.

Na rys. 1 przedstawiono przykład rachunku, jaki jest wymagany przez stronę zamawiającą oprogramowanie.



Rysunek 1. Paragon fiskalny reprezentujący wynik procesu biznesowego

2. Wymagania stawiane tworzonej aplikacji - na podstawie opisu „świata rzeczywistego” wykonanego przez eksperta problemu reprezentującego stronę zamawiającą, czyli użytkownika i klienta, przekazana wykonawcy systemu

2.1. Lista wymagań funkcjonalnych

- 2.1.1. System zawiera katalog produktów, do którego można wstawiać nowe produkty
- 2.1.2. Można zakupić trzy typy produktów różniące się sposobem obliczania ceny detalicznej: netto, z podatkiem, z promocją,
- 2.1.3. Można wprowadzić wiele rachunków o unikalnych numerach
- 2.1.4. Pozycje rachunku muszą zawierać produkty różne w sensie nazwy, ceny, podatku i promocji
- 2.1.5. Każda pozycja rachunku powinna podać swoją wartość brutto oraz dane produktu oraz ilość zakupionego produktu.
- 2.1.6. Na rachunku powinna znajdować się wartość łączna wszystkich zakupów oraz wartości zakupów należących do wybranych kategorii

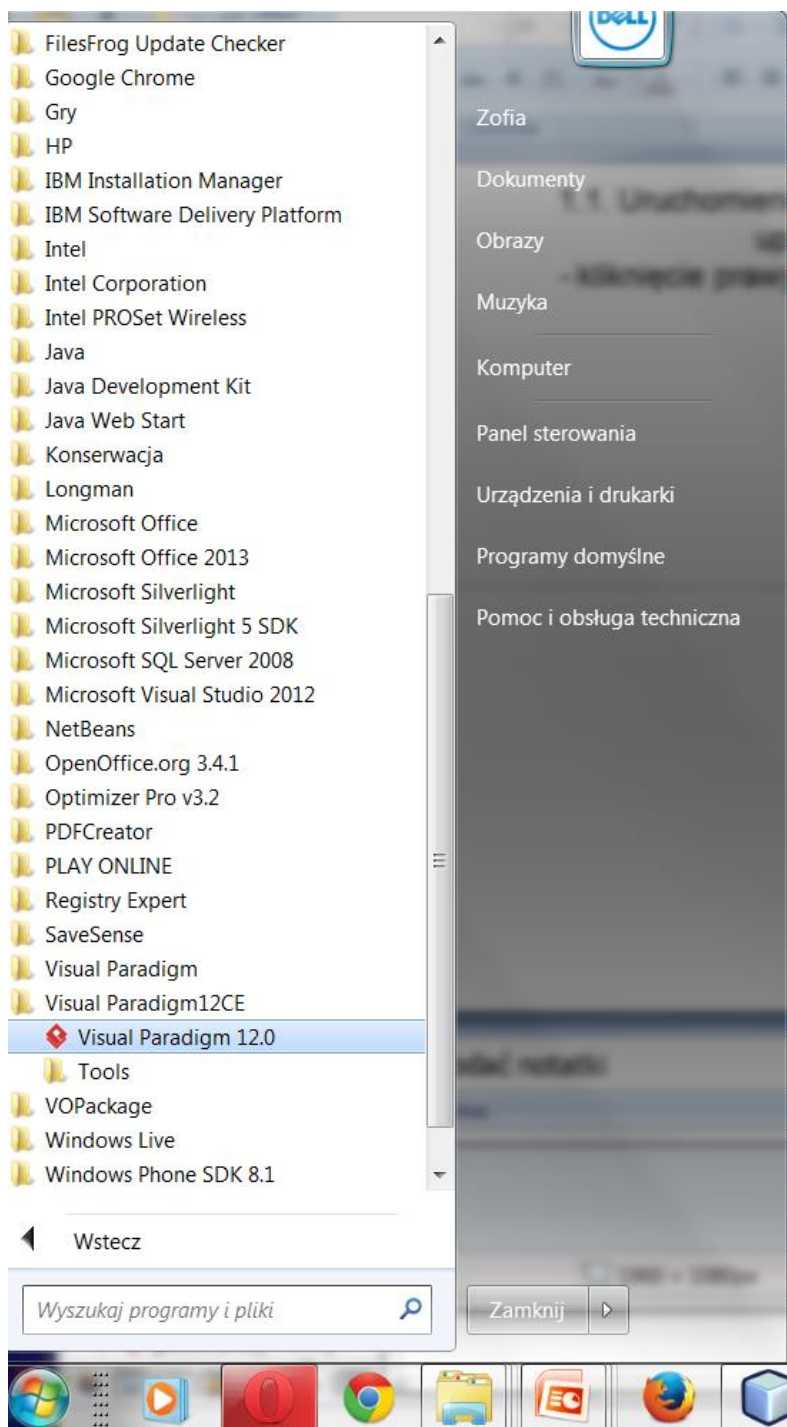
2.2. Lista wymagań niefunkcjonalnych

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

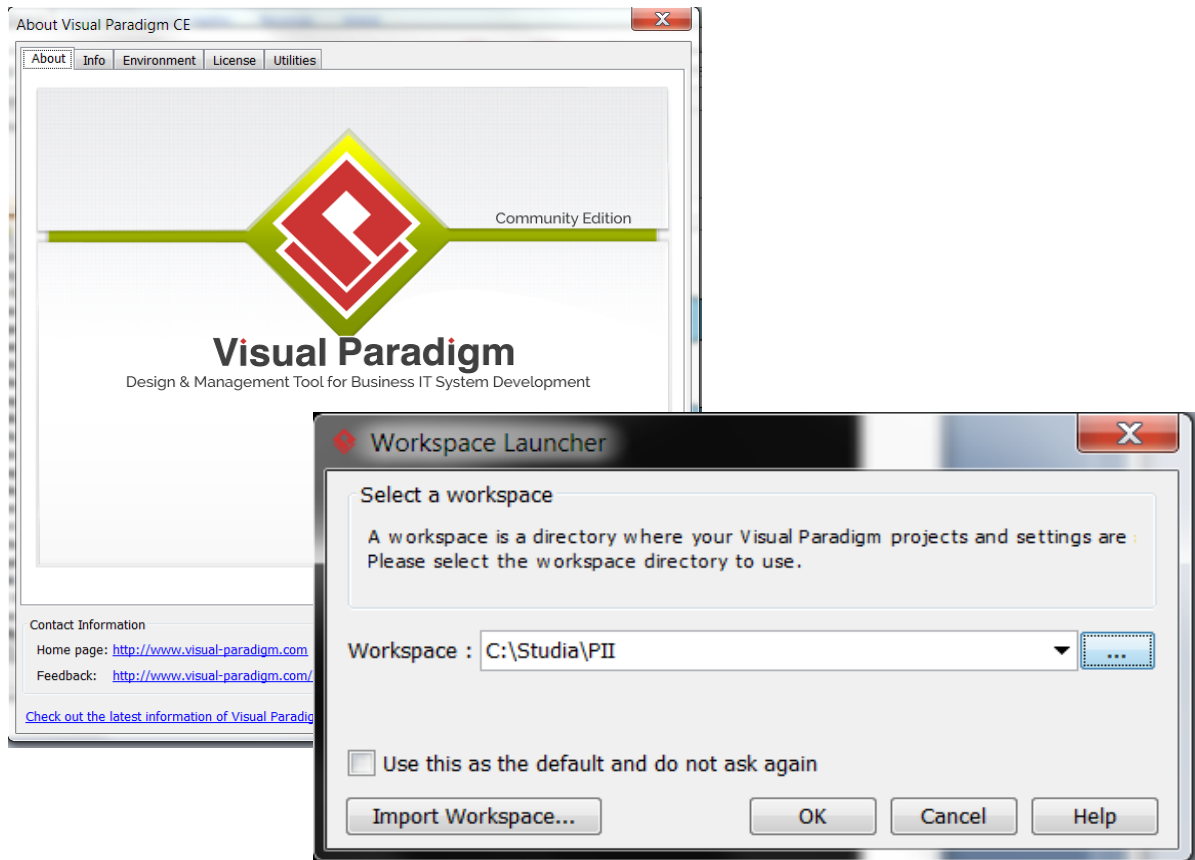
Dodatek 2

Przykład zastosowania środowiska Visual Paradigm do tworzenia opisu biznesowego „świata rzeczywistego” oraz diagramu wymagań

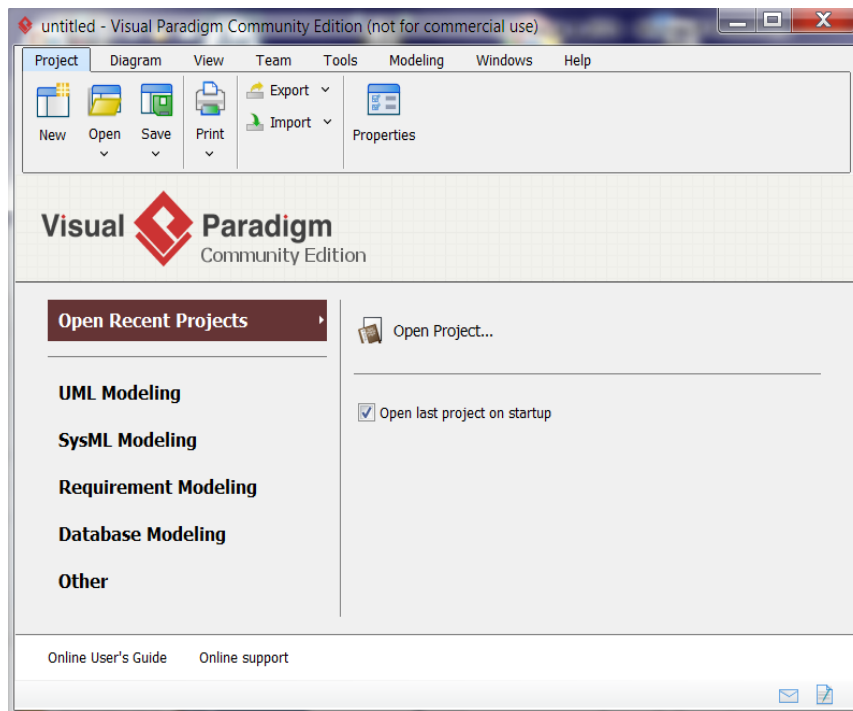
- 1. Uruchomienie programu programu *Visual Paradigm 12.0* z uprawnieniami administratora (np. kliknięcie prawym klawiszem myszy na nazwę programu).**



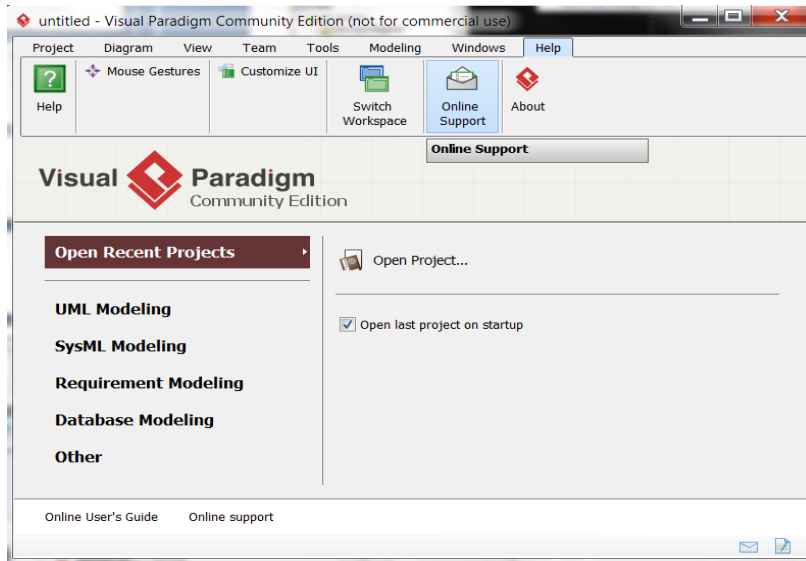
2. Wybór katalogu do tworzenia projektów.



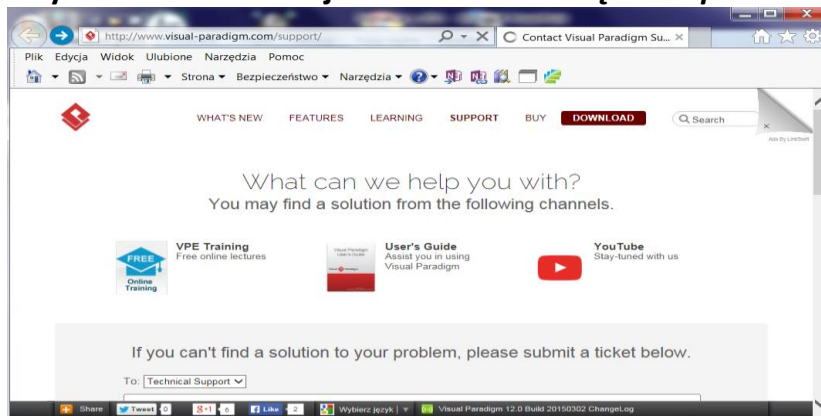
3. Start programu bez projektów w wybranej lokalizacji (workspace).



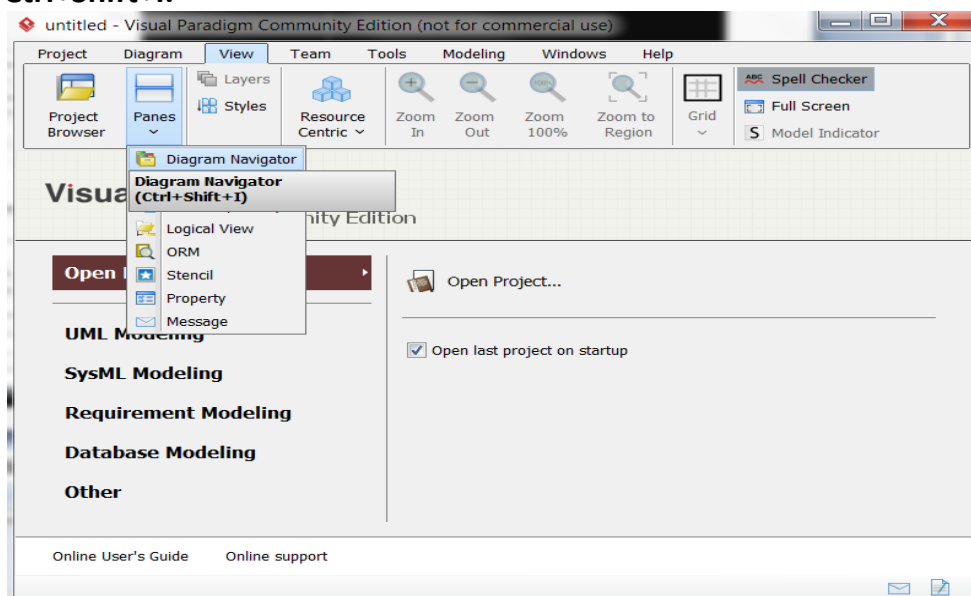
4. Korzystanie z opcji *Help* ze strony producenta narzędzia.



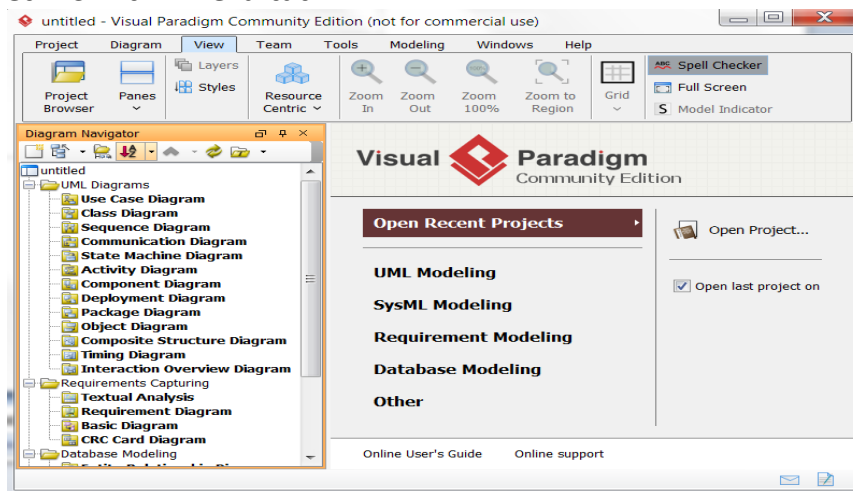
4.1. Przykładowe informacje dostarczone z części *Help*.



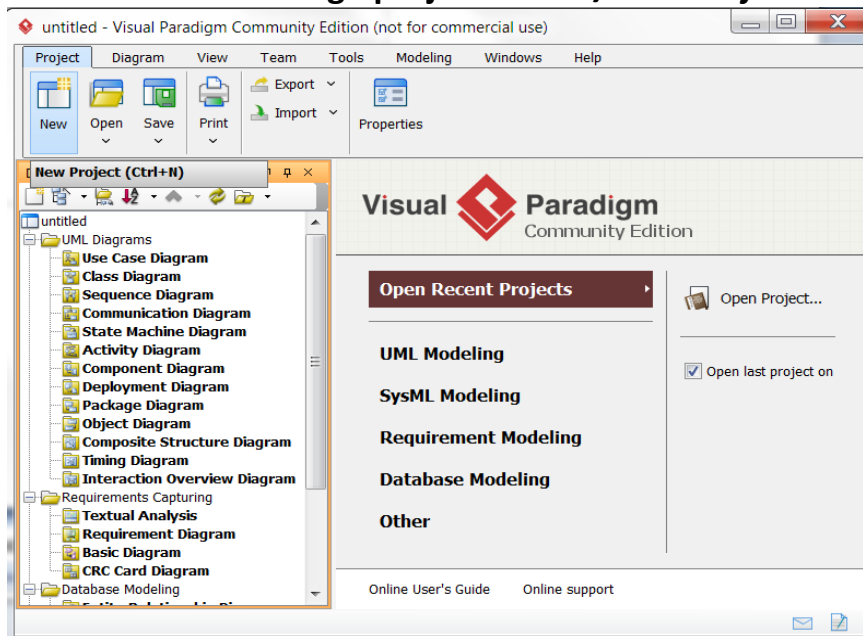
4.2. Wybór sposobu wyświetlania projektu - View/Panes/Diagram Navigator lub **Ctrl+Shift+I**.



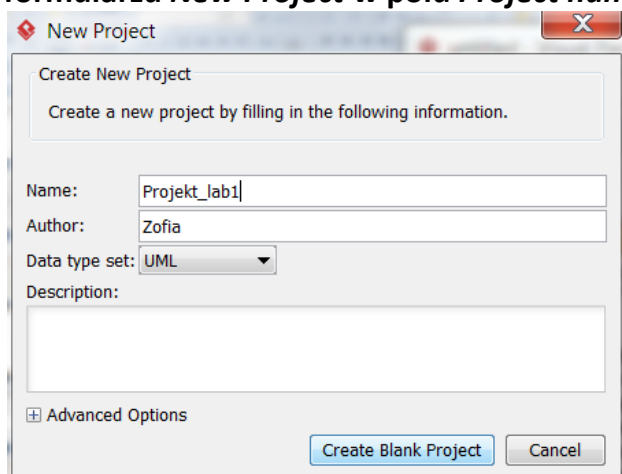
4.3. Wybór sposobu wyświetlania projektu - View/Panes/Diagram Navigator lub Ctrl+Shift+I – rezultat.



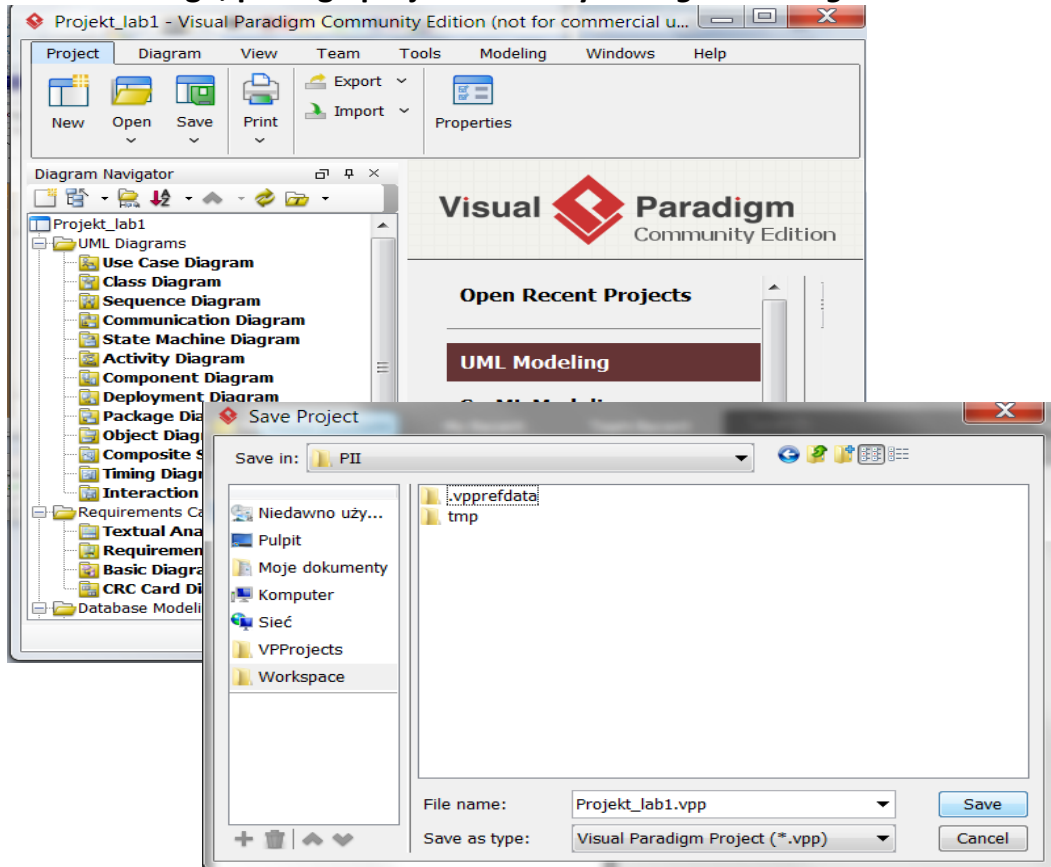
5. Utworzenie nowego projektu – File/New Project.



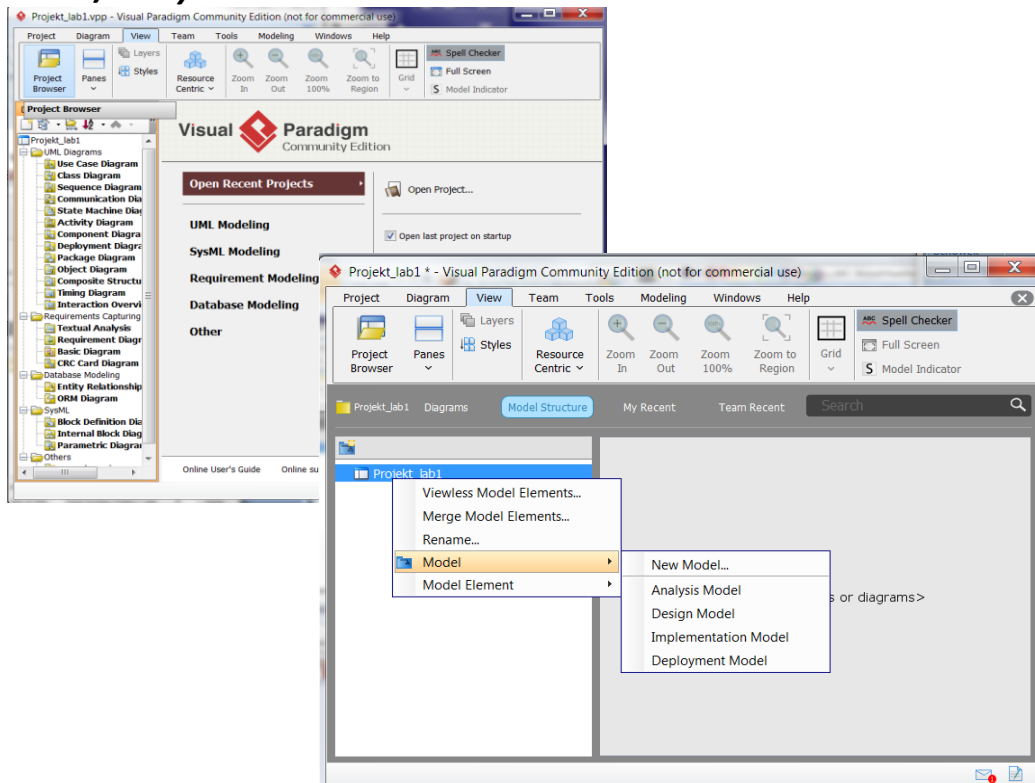
5.1. Utworzenie nowego projektu – nadanie nazwy nowemu projektowi w formularzu New Project w polu Project name.



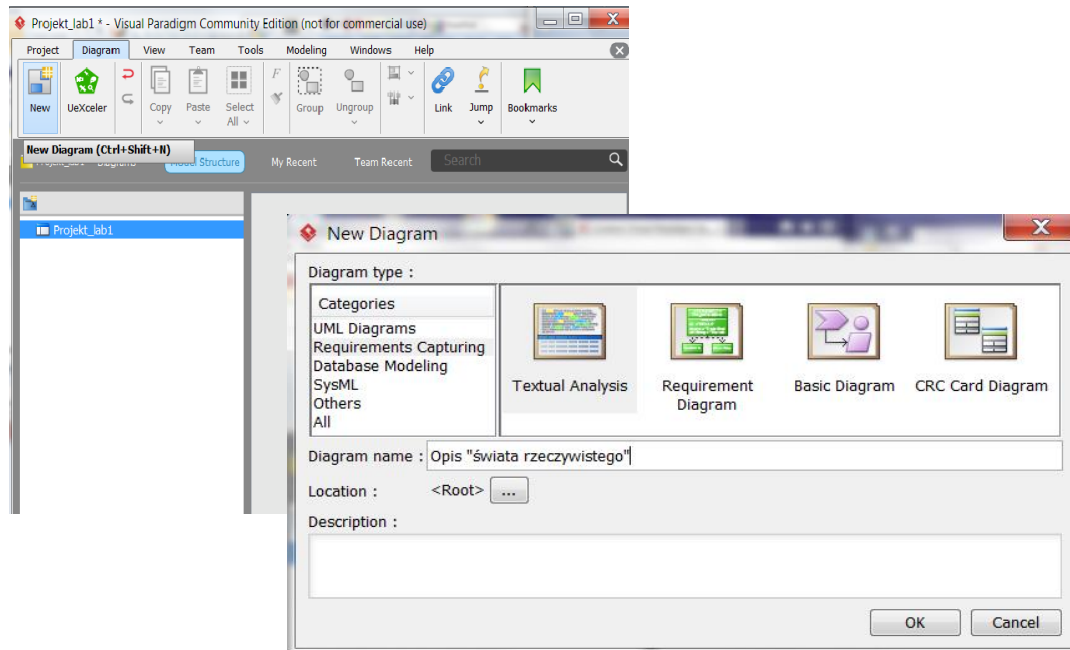
5.2. Widok nowego, pustego projektu – w stylu *Diagram Navigator*



5.3. Tworzenie modelu analizy w projekcie (w widoku *View/Project Browser*) – po kliknięciu prawym klawiszem na nazwę projektu należy wybrać z list: *Model/Analysis Model*.



6. Dodanie diagramu reprezentującego opis „świata rzeczywistego” – po kliknięciu prawym klawiszem myszy na nazwę modelu wybór z listy: *Diagram/New/Requirements Capturing/Textual Analysis*.



6.1. Należy wykonać opis biznesowy „świata rzeczywistego” – Katalog tytułów i książek w bibliotece.

1. Opis zasobów ludzkich

Pracownik wypożyczalni może dodawać do katalogu tytułów nowe tytuły. Każdy tytuł jest reprezentowany przez następujące dane: tytuł, autor, wydawnictwo, ISBN oraz informacje o liczbie egzemplarzy i miejscu ich przechowywania i występuje w bibliotece jako pojedyncza informacja dla każdego tytułu. Pewna grupa tytułów opisuje książki nagrane na kasety, dlatego dodatkowo tytuł zawiera dane nagrania np nazwisko aktora. Każdy egzemplarz, niezależnie, czy jest książką czy kasetą, jest opisany odrębną informacją zawierającą numer egzemplarza i ewentualnie (dotyczy to wyodrębnionych egzemplarzy) informację o liczbie dni, na które można wypożyczyć egzemplarz. Numery egzemplarzy mogą się powtarzać dla różnych tytułów. Pracownik biblioteki (bibliotekarz) może dodawać nowe tytuły i egzemplarze oraz je przeszukiwać, natomiast klient może jedynie przeszukiwać tytuły i sprawdzać egzemplarze wybranych tytułów.

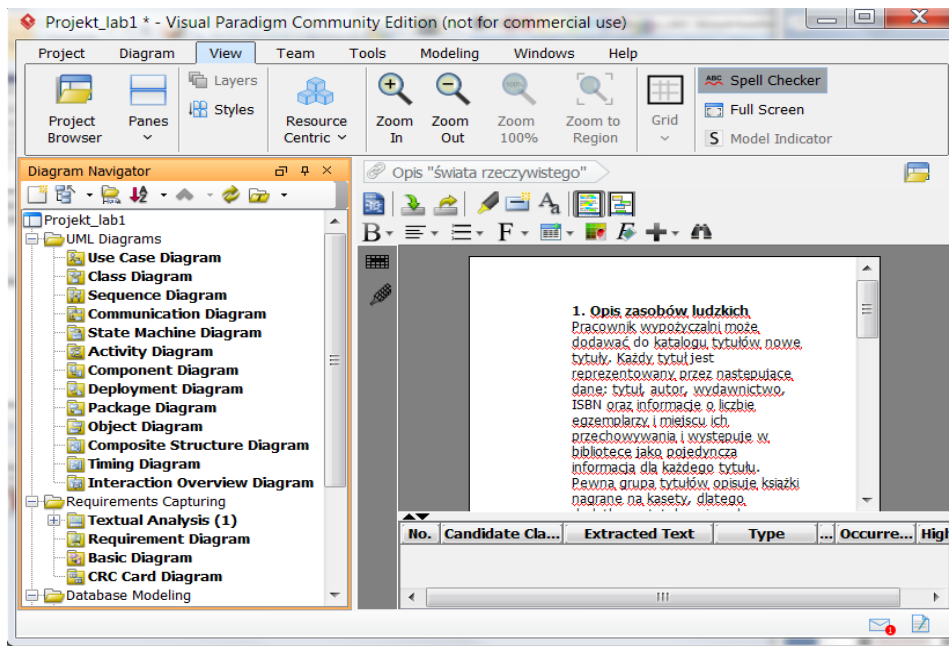
2. Przepisy

Pracownik ponosi odpowiedzialność za poprawność danych - odpowiada materialnie za niezgodność danych ze stanem wypożyczalni.

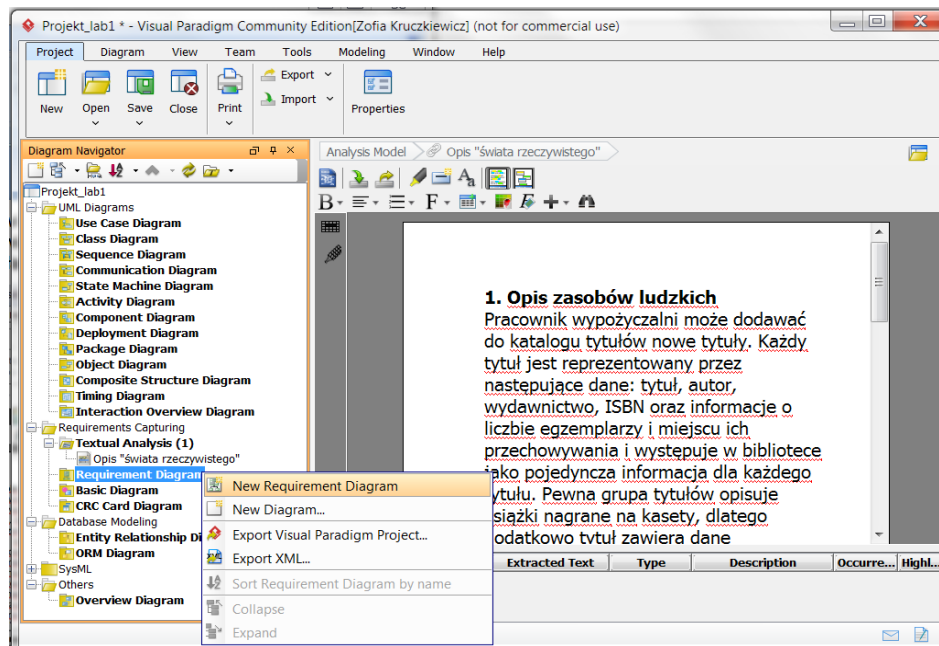
3. Dane techniczne

Klient może przeglądać dane wypożyczalni za pośrednictwem strony internetowej lub bezpośrednio za pomocą specjalnego programu. Zakłada się, że klientów jednocześnie przeglądających dane wypożyczalni może być ponad 1000 oraz wypożyczalnia może zawierać kilkadziesiąt tysięcy tytułów oraz przynajmniej dwukrotnie więcej egzemplarzy. Biblioteka składa się z kilku ośrodków w różnych miastach na terenie kraju (lista miast jest dołączona do umowy). Zaleca się stosowanie technologii Java.

6.2. Okno diagramu – należy nadać nazwę diagramowi np. Opis „świata rzeczywistego” i należy wprowadzić tekst reprezentujący opis „świata rzeczywistego” projektu (następny slajd zawiera przykład opisu).



7. Należy wyspecyfikować wymagania funkcjonalne i нефункционалне za pomocą diagramu wymagań (*Requirement Diagram*). W tym celu należy dodać ten diagram klikając prawym klawiszem myszy na pozycję *Requirement Diagram/ New Requirement Diagram*.



7.1. Należy zdefiniować wymagania aplikacji.

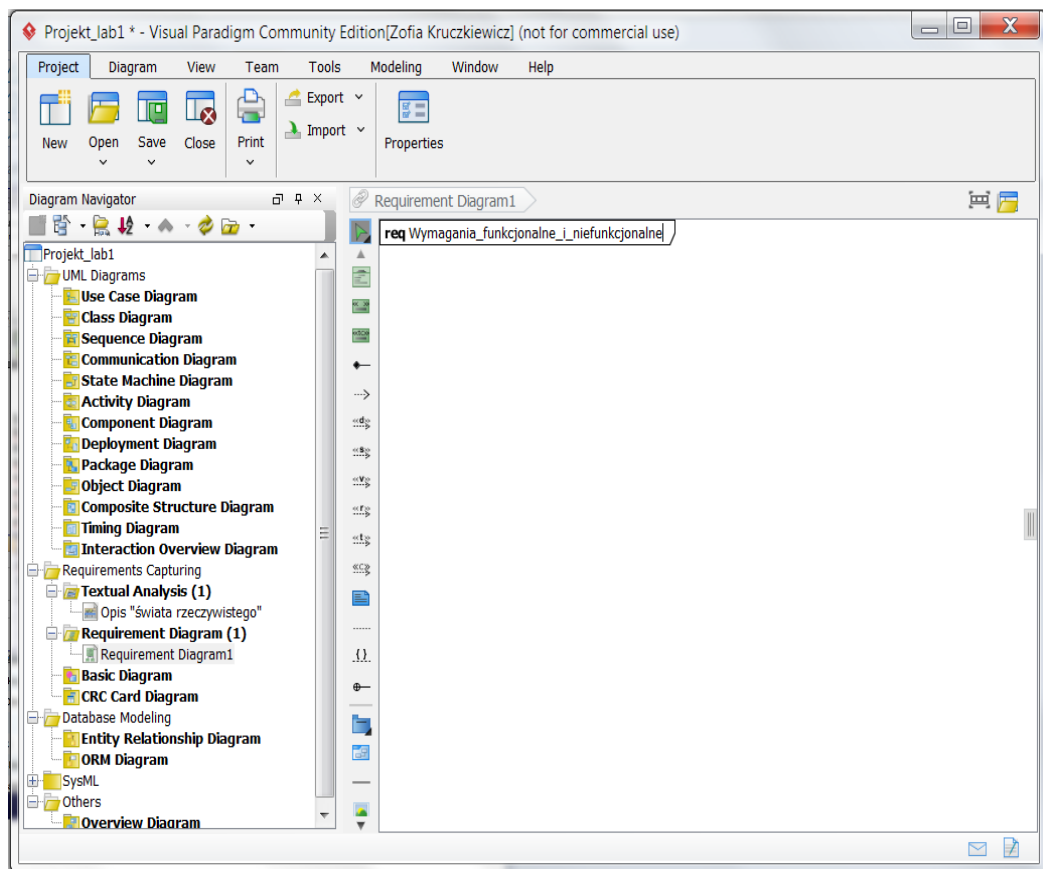
Wymagania funkcjonalne

- Biblioteka wypożycza podane książki i czasopisma osobom zarejestrowanym, o ile je posiada
- Biblioteka dokonuje zakupu nowych książek, przy czym popularne książki kupuje w kilku egzemplarzach. Usuwa zniszczone książki i czasopisma.
- Bibliotekarz jest pracownikiem biblioteki, komunikuje się z wypożyczającym. Jego praca jest wspierana za pomocą systemu
- Wypożyczający może zarezerwować książkę lub czasopismo, które nie jest dostępne w danej chwili, W momencie, kiedy zamówione rzeczy są dostępne- albo po zwrocie lub dzięki zakupowi, można je wypożyczyć i usunąć rezerwację. Rezerwację można usunąć niezależnie.
- Biblioteka może łatwo utworzyć, zmienić i usunąć informację o tytułach, wypożyczających, wypożyczeniach i rezerwacjach

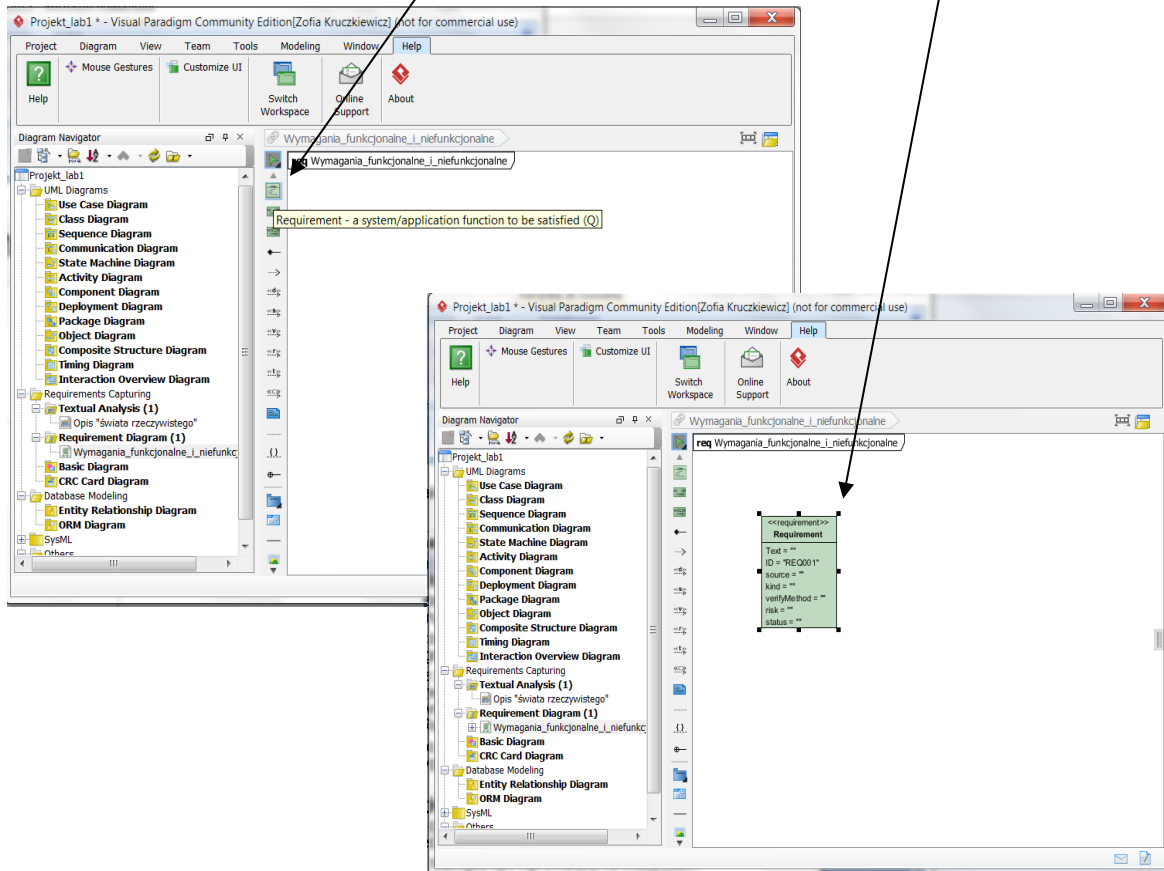
Wymagania niefunkcjonalne

- System powinien pracować w popularnych systemach (UNIX, Windows, OS/2) i powinien mieć nowoczesny graficzny interfejs użytkownika
- System powinien się rozwijać np. wprowadzenie możliwości zawiadamiania rezerwującego książkę o jej dostępności

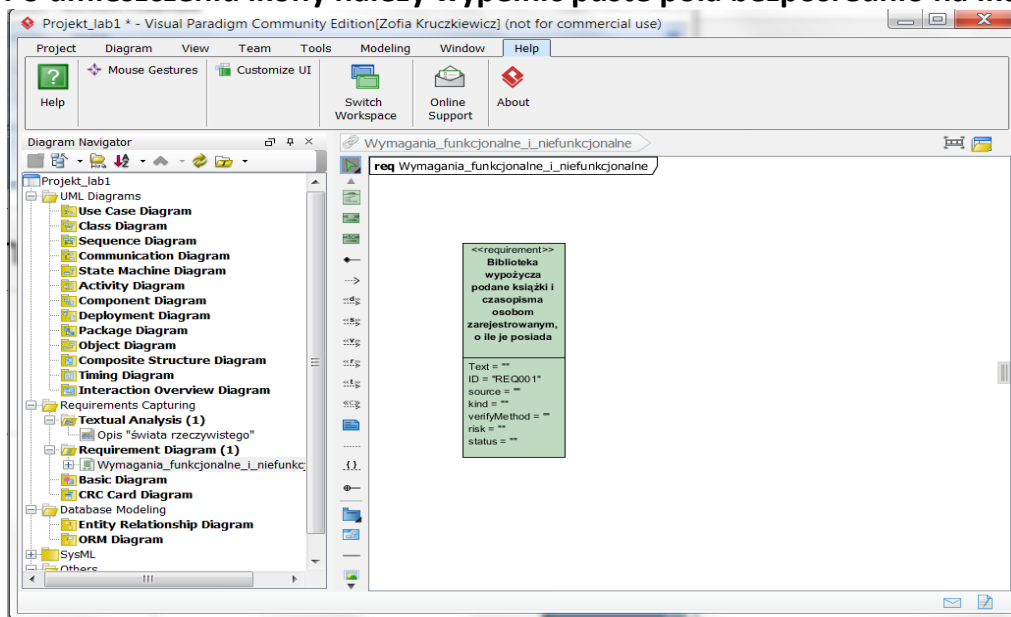
7.2. Należy wpisać nazwę diagramu wymagań w lewym górnym rogu diagramu.



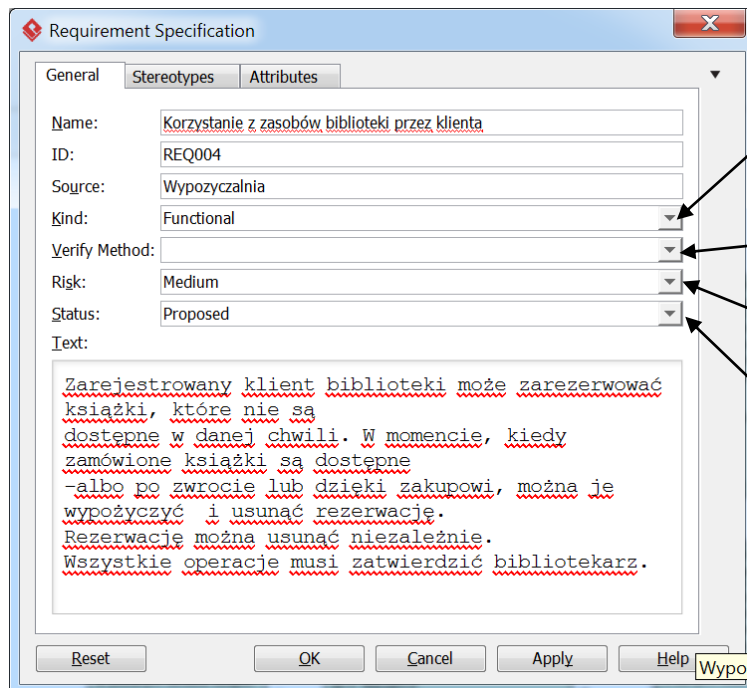
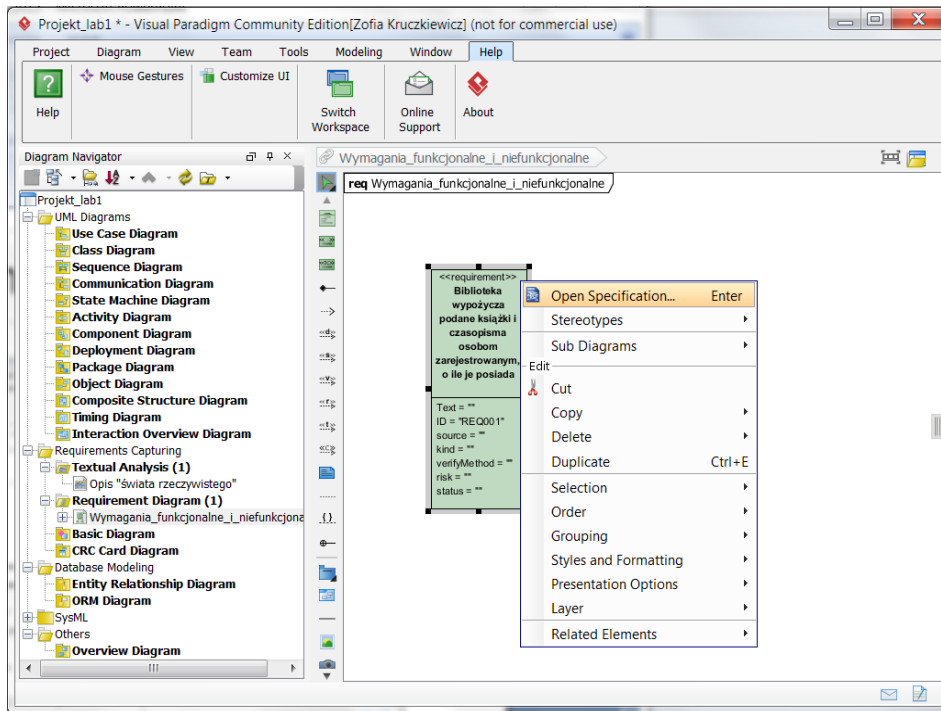
7.3. Należy umieścić na powierzchni diagramu ikony wymagań funkcjonalnych przeciągając je z palety umieszczonej z lewej strony diagramu i upuszczając na powierzchni diagramu



7.4. Po umieszczeniu ikony należy wypełnić puste pola bezpośrednio na ikonie



lub po kliknięciu prawym klawiszem myszy wykonać to za pomocą wybranego formularza *Open Specification/Requirement Specification*.



- Functional
- Functional
- Performance
- Interface
- Proposed
- Analysis
- Demonstration
- Inspection
- Test
- Medium
- High
- Medium
- Low
- Proposed
- Proposed
- Approved
- Rejected
- Deferred
- Implemented
- Mandatory
- Obsolete

7.5. Rezultat – wykonano powiązania pomiędzy poszczególnymi wymaganiami za pomocą relacji Composition, Dependency oraz Derive (<<derive>>) przeciągając je z palety z lewej strony diagramu, następnie upuszczając je na powierzchnię diagramu i łącząc nimi poszczególne wymagania. Diagram wymagań funkcjonalnych – 1-y i 2-i fragment

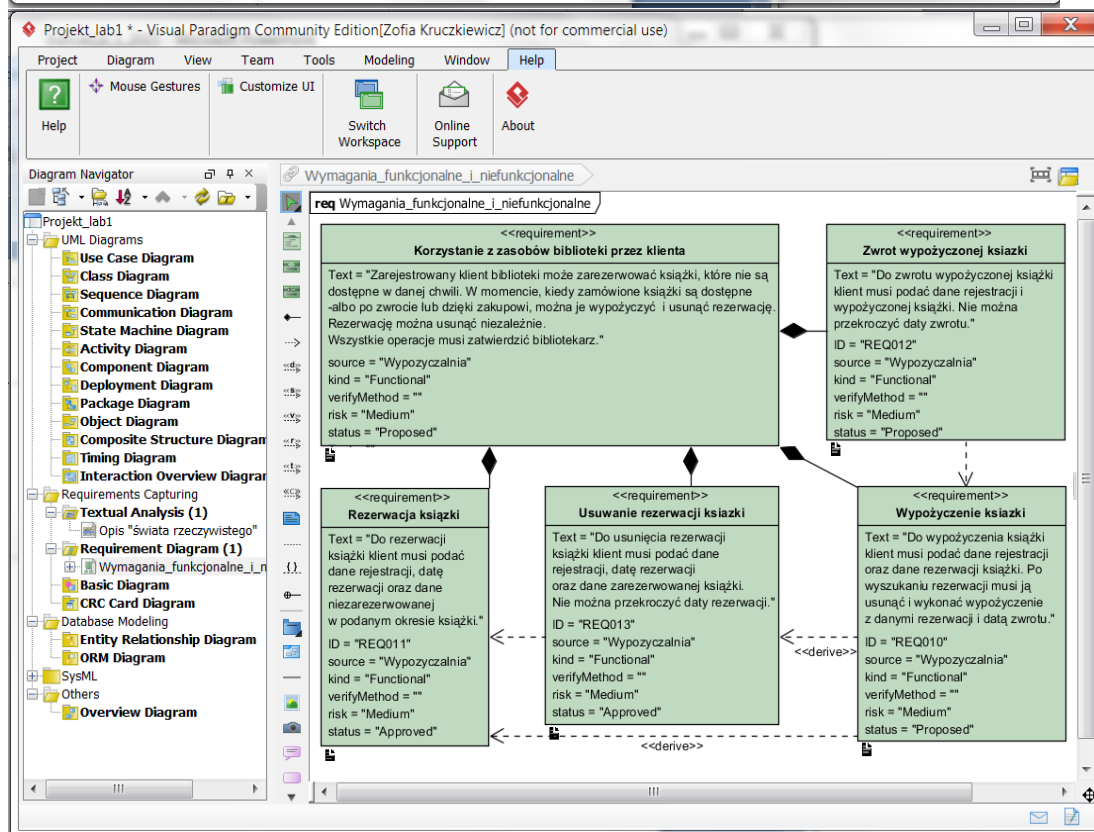
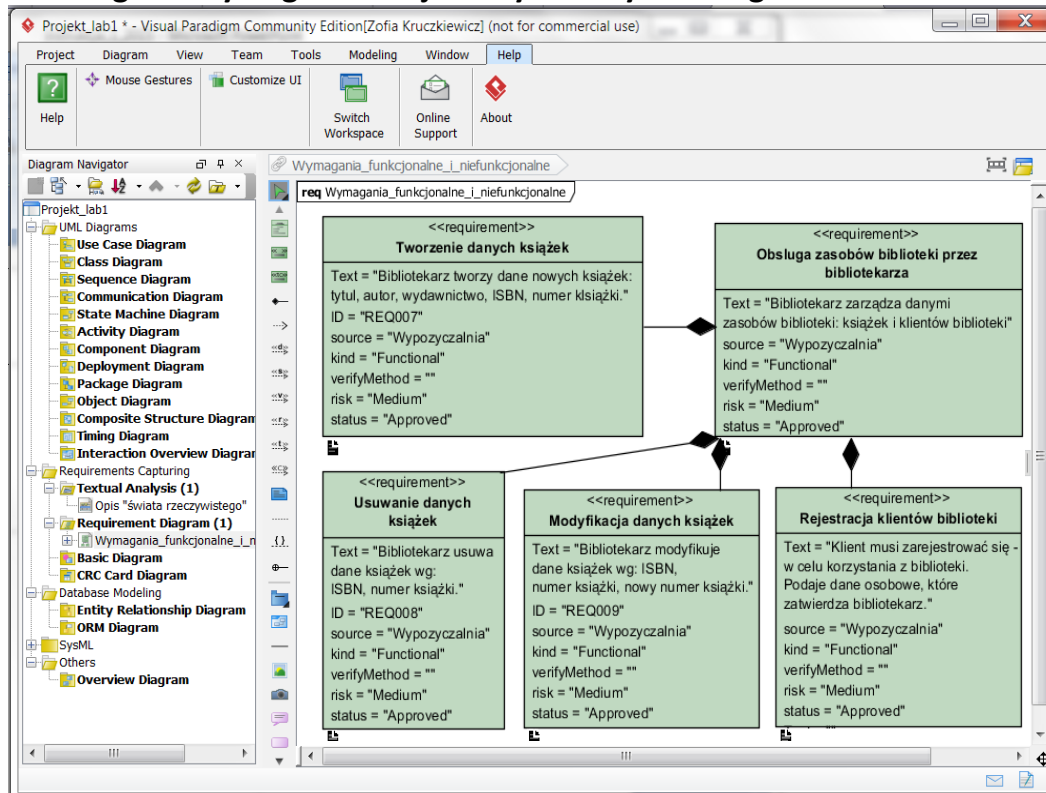
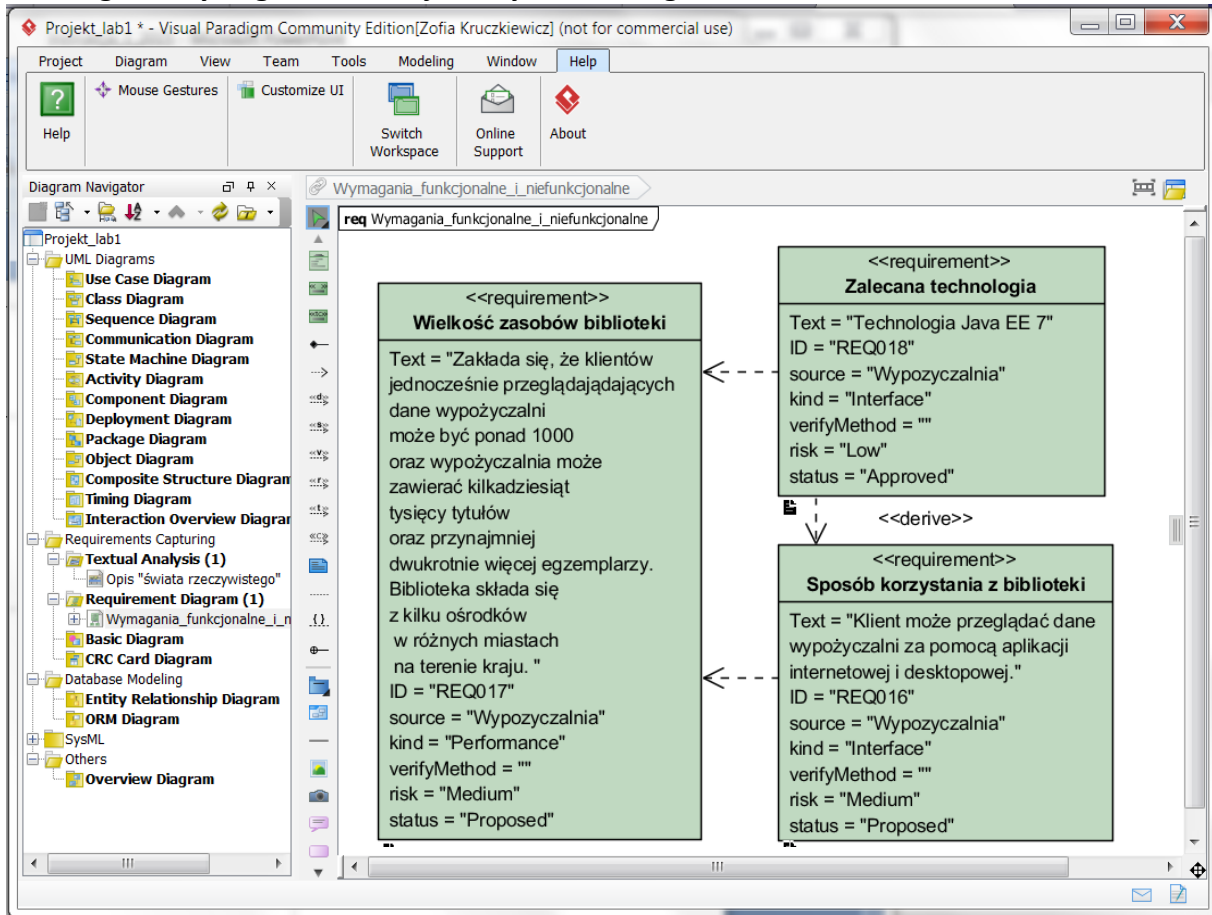


Diagram wymagań niefunkcyjnych – 3-i fragment



Uwaga: Można wykonać trzy diagramy wymagań

8. Dodatkowa pomoc ze strony Visual Paradigm:

Pomoc: [Creating requirement diagram.](http://www.visual-paradigm.com/support/documents/vpuserguide/94/158/6516_creatingrequ.html)

(http://www.visual-paradigm.com/support/documents/vpuserguide/94/158/6516_creatingrequ.html)