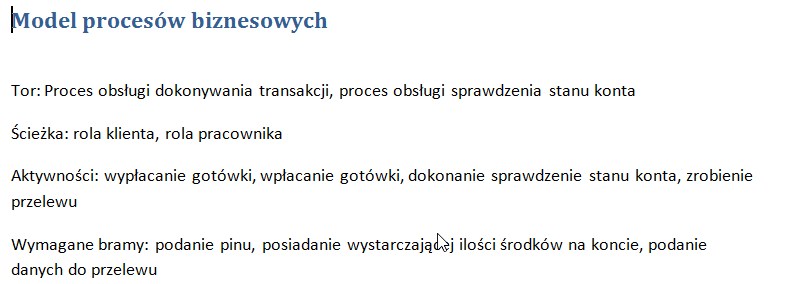
**1 Model procesów biznesowych**

**Aktywność – działanie w systemie np. nauczanie**

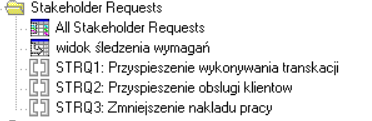
**Brama – warunek np. trzeba się stawić o określonej godzinie**

**Ścieżka – role, cechy biznesu jakie definiują role np. rola studenta, wykładowcy**

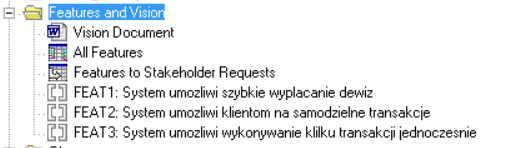
**Tor np. nazwa procesu np. udzielanie konsultacji**

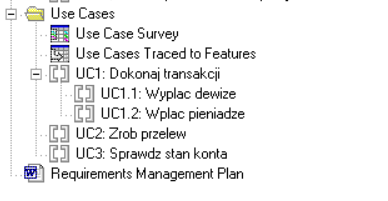
****

**2. STRQ**

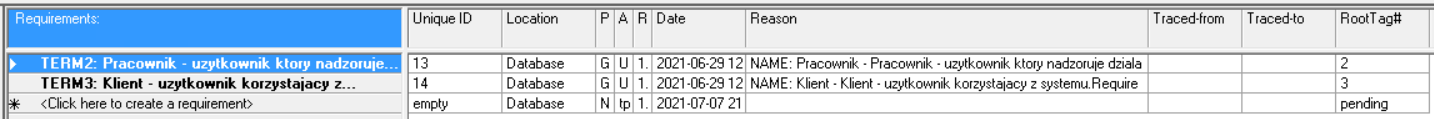
****

**3. Widok wymagań FEAT i UC**

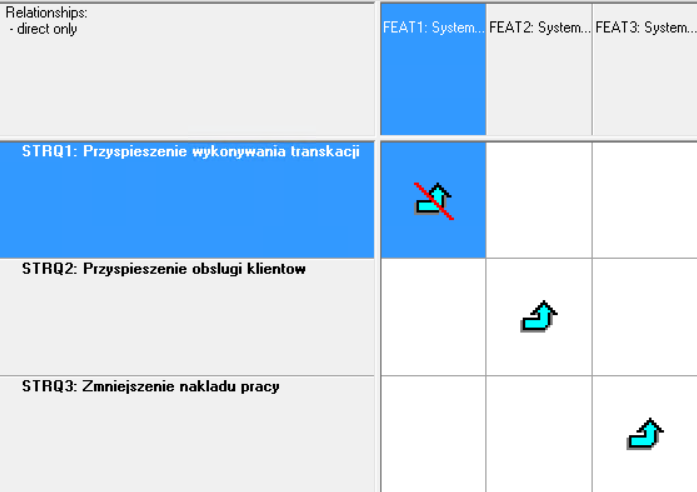
****

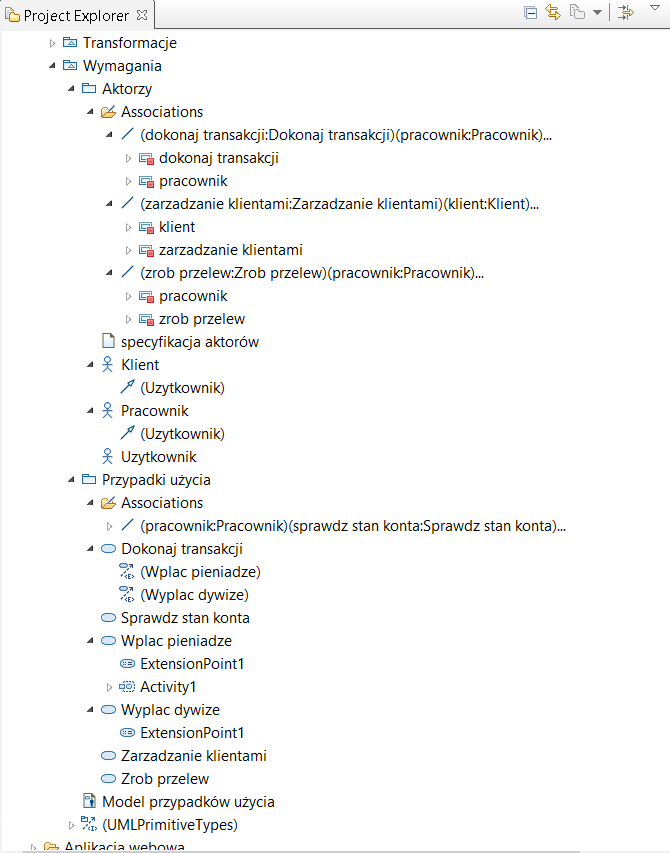
****

**4. Wymagania TERM**

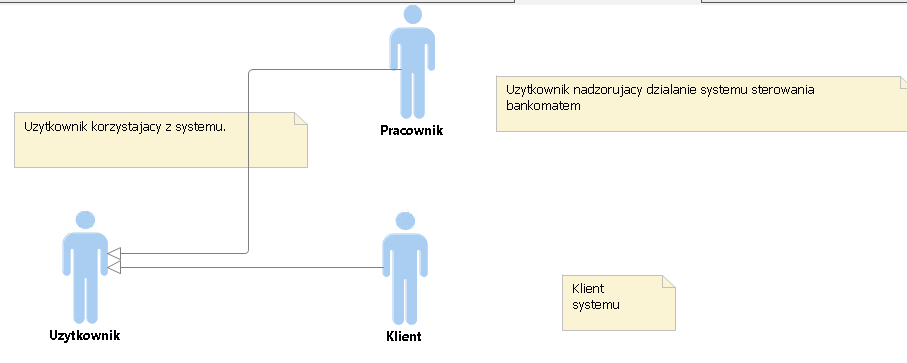
****

**5. Widok śledzenia wymagań STRQ na FEAT**

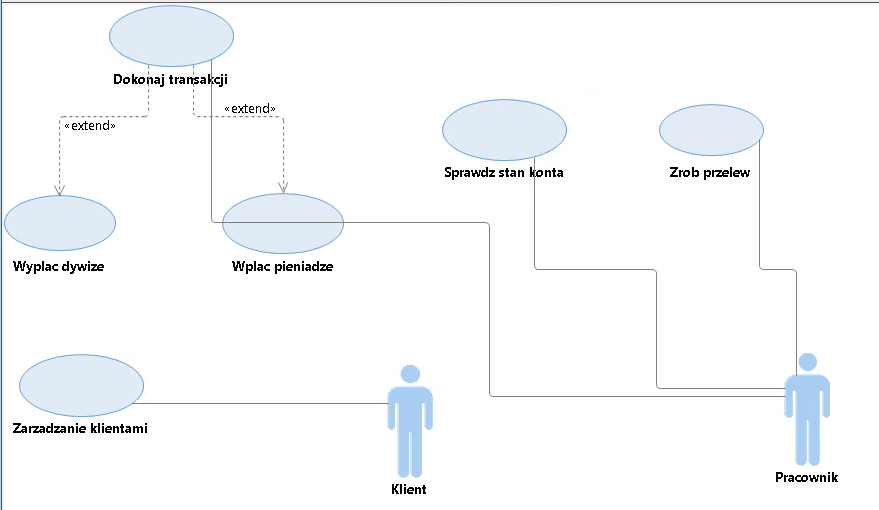
****

****

**6. Specyfikacja aktorów**

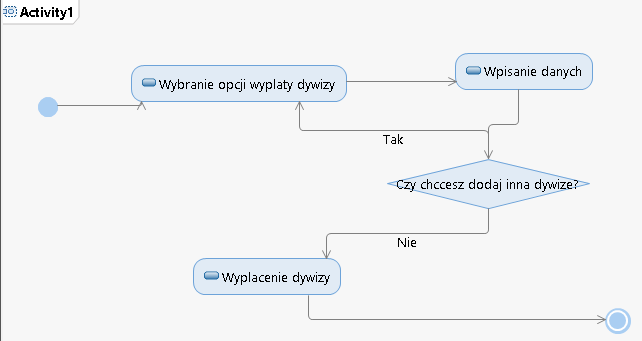
****

**7. Model przypadków użycia (ze struktulizacją UC)**

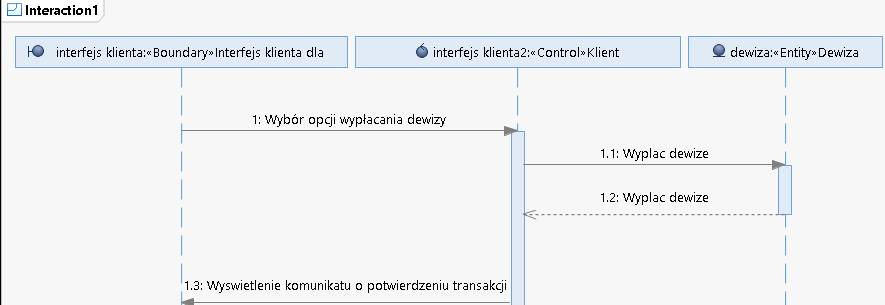
****

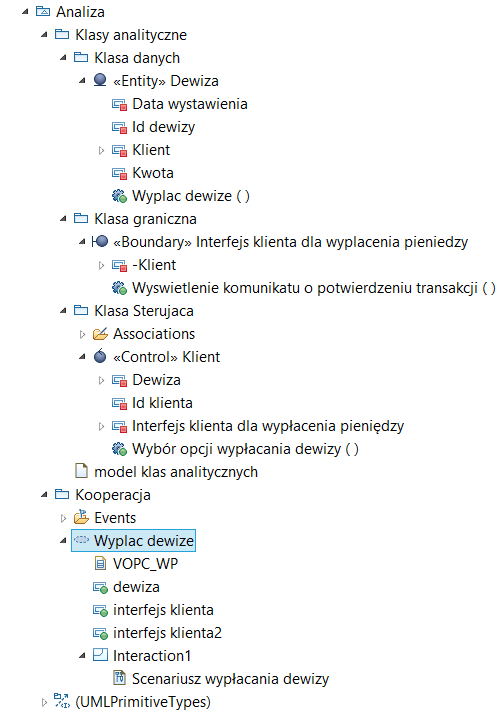
**8. Scenariusz na wybranego przypadku użycia – diagram aktywności. Stasiak pod danym use case stworzył ten diagram.**

Activity1 jak coś pod danym use case

****

9. Model realizacji przypadków użycia - diagram sekwencji dla danego use case. Stasiak po prostu je przekleił i zostawił ten jeden wybrany .S skopiował folder Przyp.uż i zmienił mu nazwę na kooperacja. Zaznaczam use case ten jeden z kooperacji ->refactor->convert to new Element -> class Type->collaboration



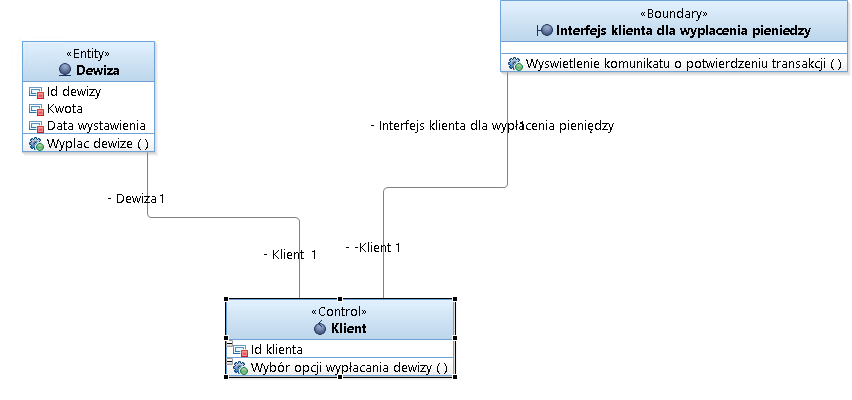


10. Specyfikacja klas analitycznych No to specyfikacja cóż mówić

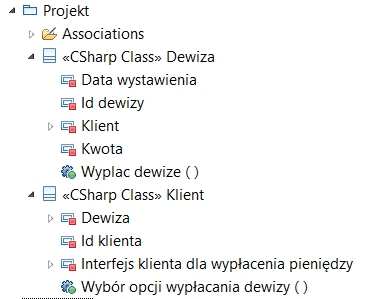
Dajemy no całą analizę albo paczkę klasy analityczne – prawym – preferences -profiles i dodajemy rup. A same klasy to analogicznie ale wybieramy stereotyp i dodajemy atrybuty. Jedna klasa control i przynajmniej jedna entity.

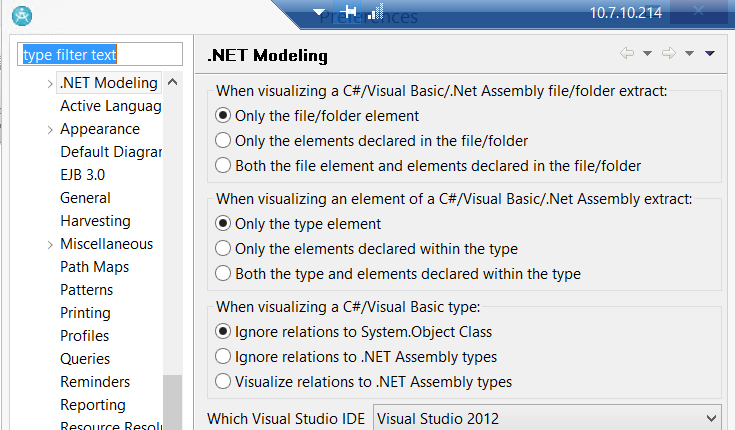
11. diagram VOPC realizacji wybranego przypadku użycia

Co do diagramu to diagram klass a drugi to diagram sekwencji/interakcji.



12. Modele klas projektowych (warstwy sterującej i danych)



Kopiujemy klasy i dajemy typ c#. Następnie nowy package transofrmacje. Otwieramy preferences w jazzie i ustawiamy 

Dalej Vs – nowy projekt-koniecznie .net i patrzymy na język ma być c# nie Visual Basic. Dalej zapisujemy projekt, odpalamy. Nowa transformacja jako source dajemy nasz klasę klas c# a jako żródło (to po prawej) dajemy ścieżkę do naszej aplikacji plik sln. I dalej, zaznaczamy tam opcje dalej i koniec. Run i koniec .

13. Transformacje dla inżynierii wprzód (do generacji kodu c# i SQL)

14. Diagramy logicznego i fizycznego modelu danych

15. Skrypt SQL

16. Kod aplikacji webowej na podstawie warstwy sterowania