

**Skład grupy:**

1.Łukasz Karbowski

2.Dawid Gramacki

Opis projektu:

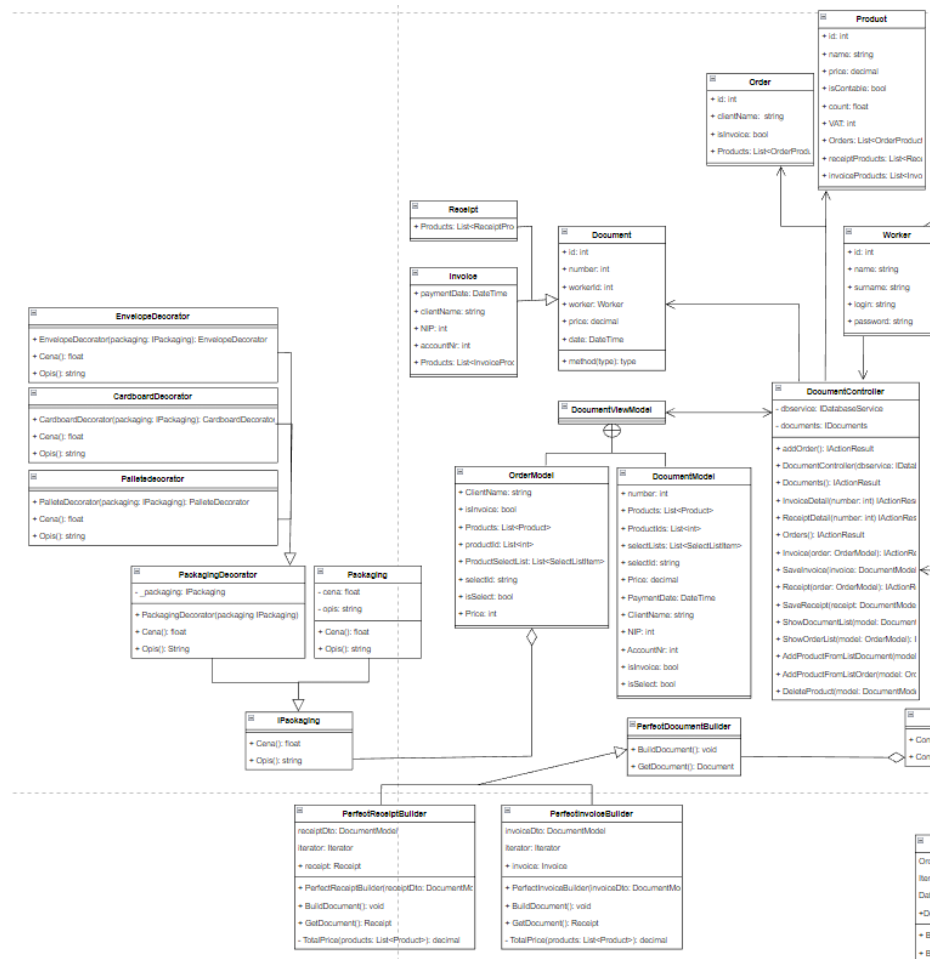
Będzie to program realizowany w technologii MVC w języku C# do obsługi zamówień pewnej firmy.

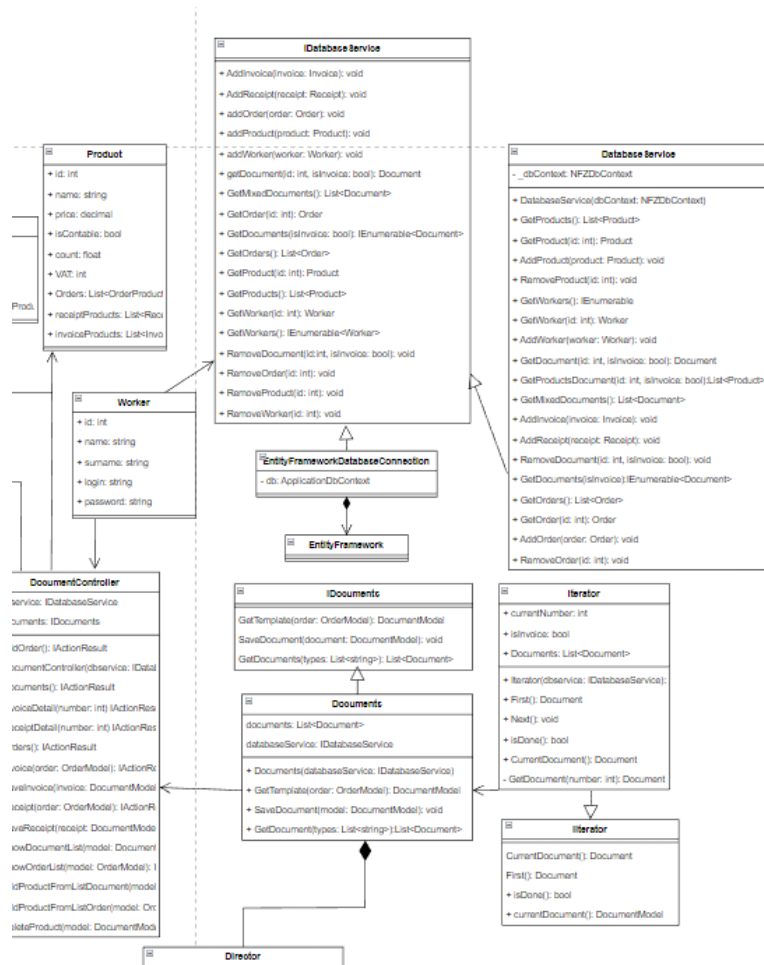
Funkcjonalności:

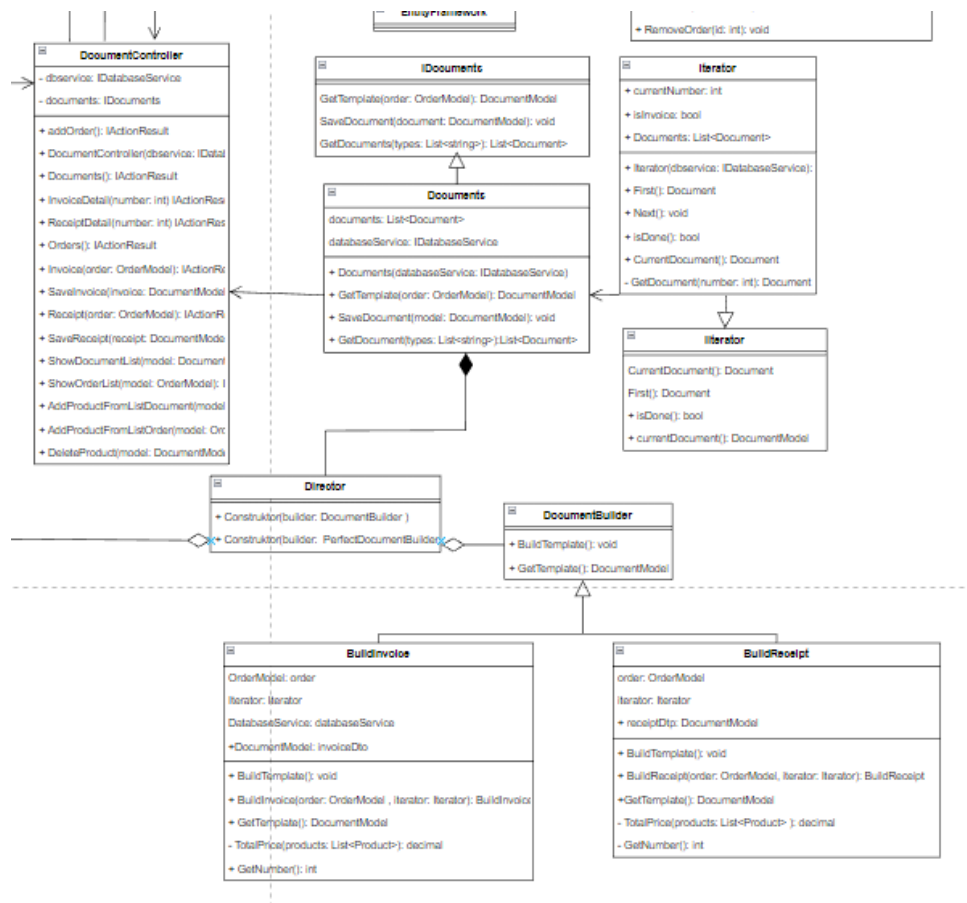
- wprowadzanie towaru do magazynu (nazwa, ilość, cena, stawka VAT)
- składanie zamówień (lista towarów ich ilość, kto składa zamówienie)
- realizacja zamówień, wystawianie faktur
- wydawanie towaru (pomniejszenie ilości magazynowych)

Projekt jest za długi dlatego załączam link do githuba: <https://github.com/Lethalman777/NFZ>

Diagram klas:







Krótki opis każdego wzorca:

Facade:

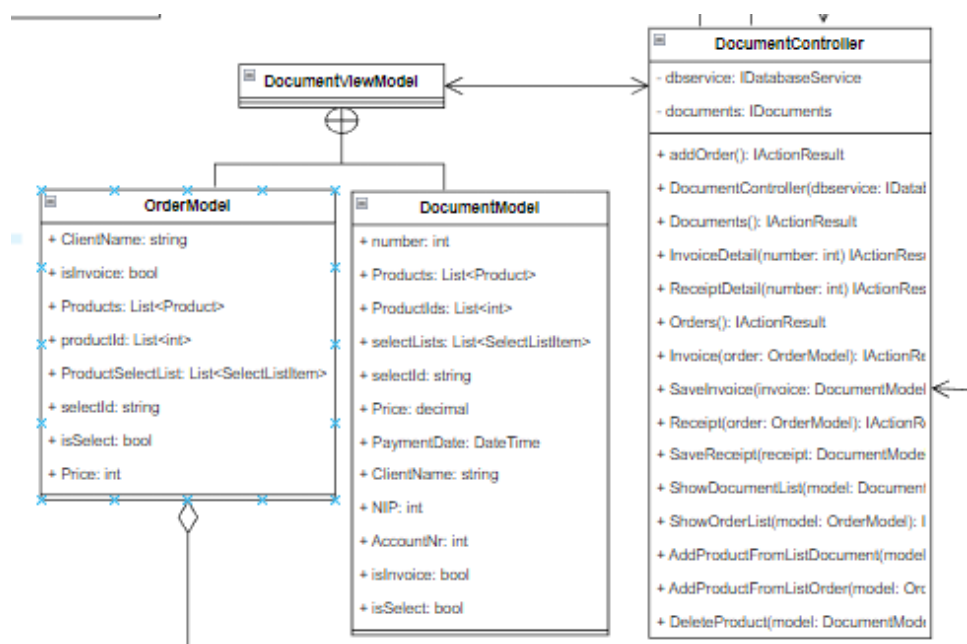
Rolą fasady jest udostępnienie interfejsu do obsługi bazy danych. Przygotowana fasada oferuje niezbędne metody do poprawnej komunikacji między aplikacjami a bazą danych.

Wektorem zmian, jest dodanie więcej metod do komunikacji gdy baza danych rozszerzy nam się o kolejne tabele.



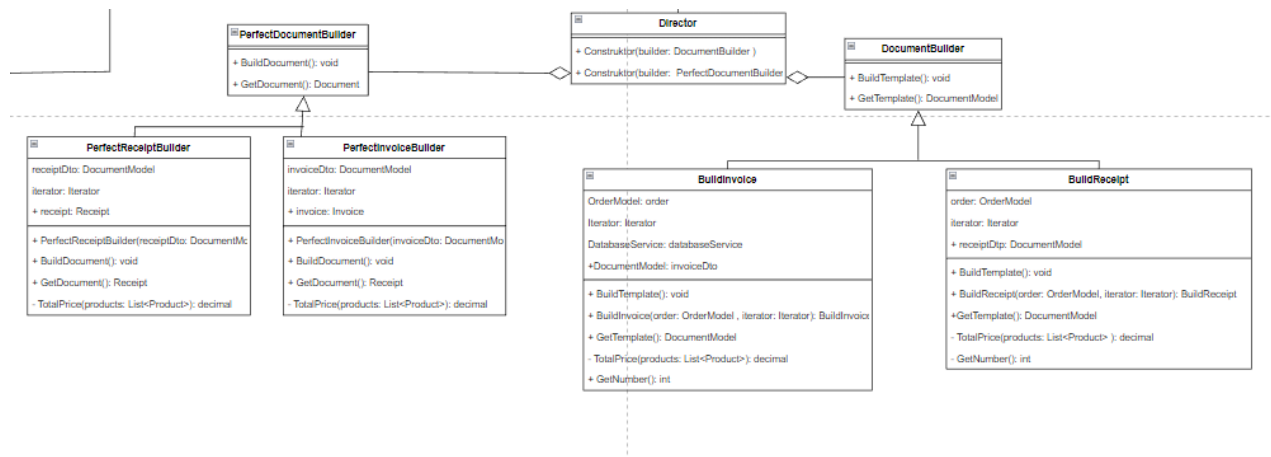
MVC:

Wzorcem architekuralnym w naszym projekcie jest wzorzec model, view, controller. Pozwala on oddzielić model od widoku co pomaga usprawnić aplikację.



Builder:

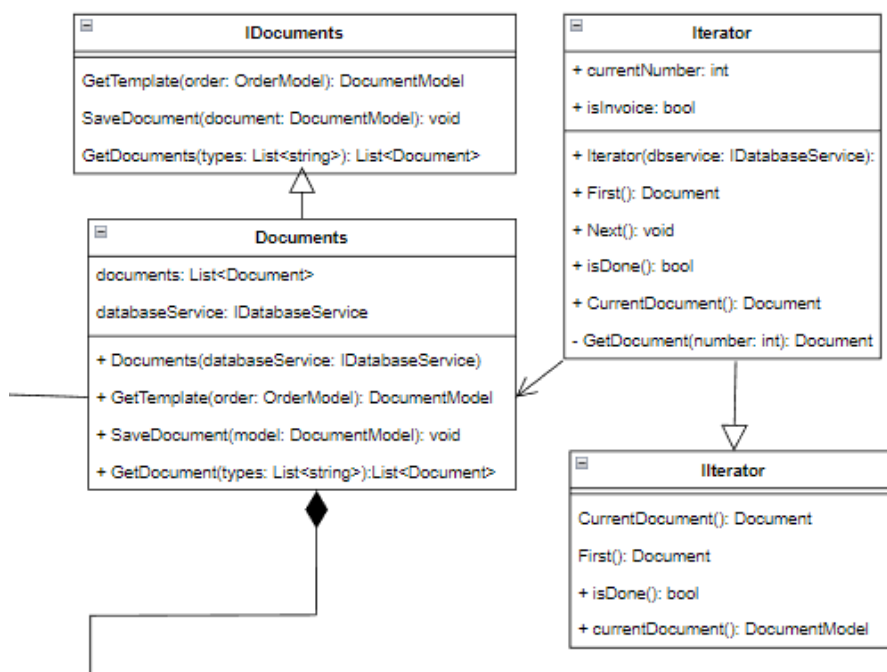
Wzorec Builder został wykorzystany w celu tworzenia dokumentów. Dyrektor posiada algorytm, który w zależności od potrzeby tworzenia faktury lub paragonu tworzy obiekt danej klasy.



Iterator:

Iterator służy do poruszania się po wyświetlonej liście dokumentów.

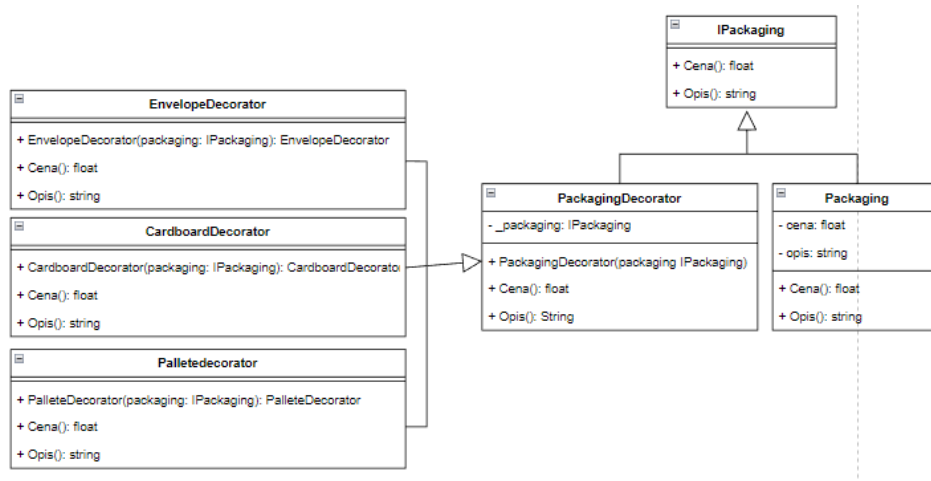
W przyszłości kolejne rzeczy również mogą być obsługiwane przez iteraotr (można dać mu więcej pracy).



Decorator:

Dekorator ma za zadanie dodanie do naszego dokumentu typu opakowania wysyłki oraz podniesienie kwoty związanej z tą właśnie fakturą/paragonem.

Dzięki niemu będzie można rozbudować aplikację w przyszłości np. O wpisanie do zamówienia różnych środków transportu lub różnych metod płatności także za pomocą dekoratora.



Opis co najmniej jednego rozwiązania specyficznego dla użytej technologii i podać wykorzystane źródło wiedzy na ten temat:

Fasad jest to wzorec strukturalny, który doskonale wpasowuje się w projekt dostarcza on przede wszystkim przejrzystość. Odpowiednia agregacja poleceń pozwala na posiadanie czytelnego interfejsu dla złożonej o wielu "kroków" logiki. Dzięki fasadzie ujednolicamy dostęp do złożonego systemu poprzez wstawienie uproszczonego uporządkowanego interfejsu programistycznego, który ułatwia jego użycie.

Źródło:

<https://blog.orbisbit.com/wpisy/wzorec-fasada-facade>

[https://pl.wikipedia.org/wiki/Fasada_\(wzorec_projektowy\)](https://pl.wikipedia.org/wiki/Fasada_(wzorec_projektowy))

Opis podziału pracy w zespole:

Łukasz Karbowski: Fasada, Builder, MVC, Iterator, diagram UML, dokumentacja, html, komentowanie kodu

Dawid Gramacki: Decorator, MVC, diagram UML, poprawienie diagramu UML, dokumentacja, html, css, komentowanie kodu

Krótką instrukcją użytkownika:

System do obsługi dokumentów, możemy wystawiać faktury lub paragony dla konkretnego zamówienie, wybierać rodzaj przesyłki oraz zakończyć transakcję. Ilości magazynowe automatycznie się zmniejszają po wystawieniu dokumentu. System dostaje zamówienie a my na jego podstawie o

ile produkt jest w magazynie wystawiamy fakturę lub paragon nanosimy poprawki, oraz po wystawieniu odejmujemy towar z magazynu.

Instrukcja instalacji:

Jest to aplikacja webowa, więc wystarczy przeglądarka.