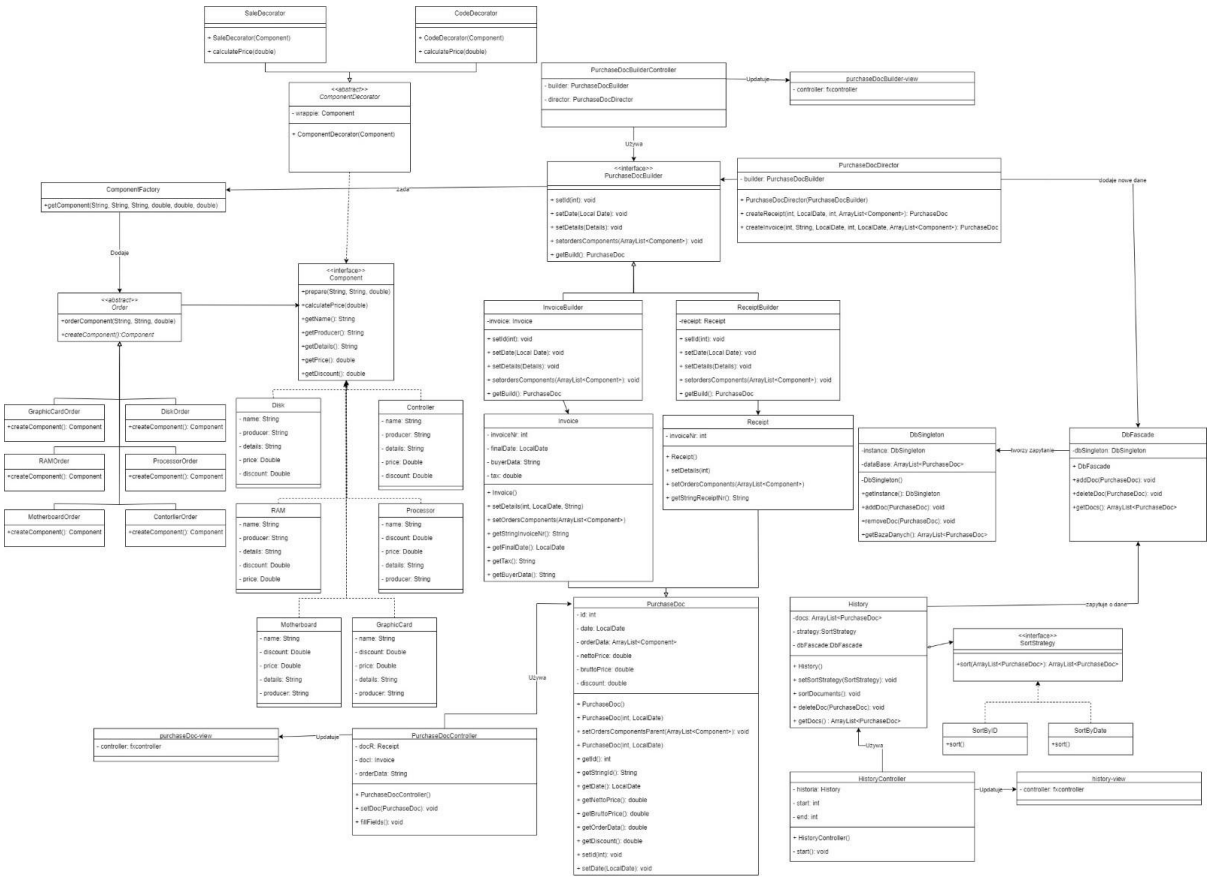
Opis i cel projektu

Tematem naszego projektu jest aplikacja, która będzie wspierała zarządzanie biznesem (obszar dokumentacji zamówień) w dziedzinie składania komputerów stacjonarnych. Aplikacja będzie umożliwiała m.in. tworzenie dokumentów sprzedaży, określanie dokładnych danych na temat zamówień, przeglądanie historii zamówień(dokumentów). Aplikacja jest przeznaczona dla małych przedsiębiorstw.

Opis funkcjonalności

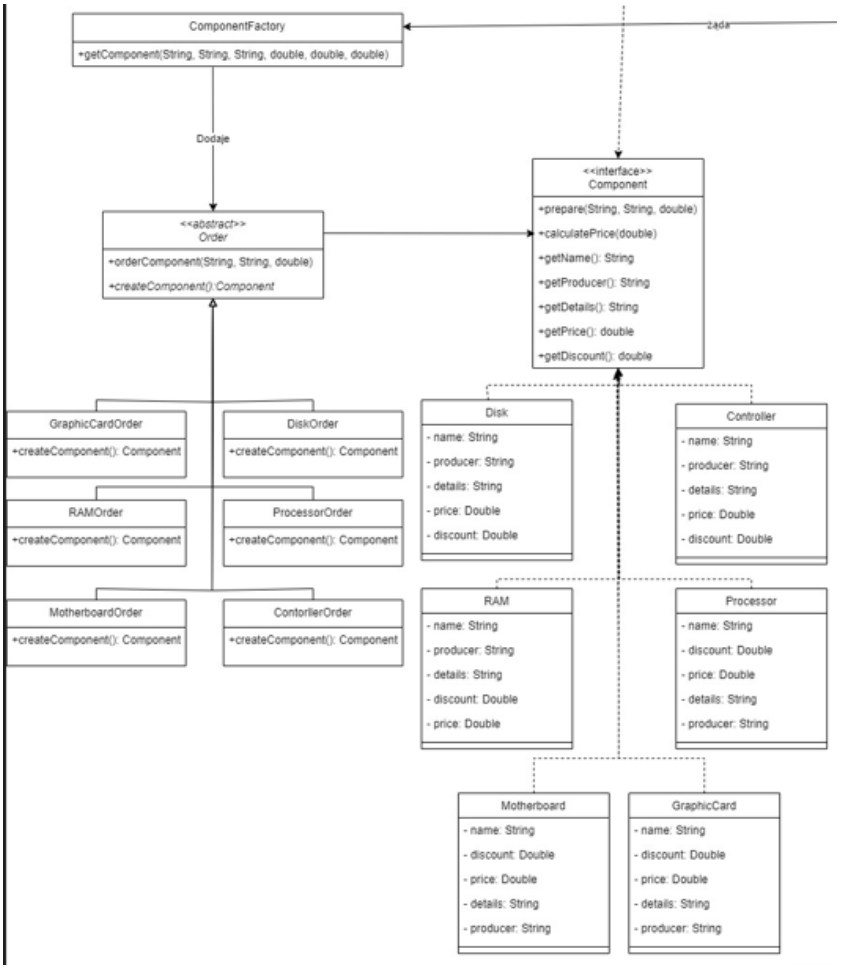
1. Dostęp do bazy przechowującej dane na temat zamówień i ich historii
2. Utworzenie dokumentu potwierdzającego dokonania zakupu(faktura/paragon)
3. Sortowanie dokumentów
4. Podgląd szczegółów w dokumencie

Diagram klas



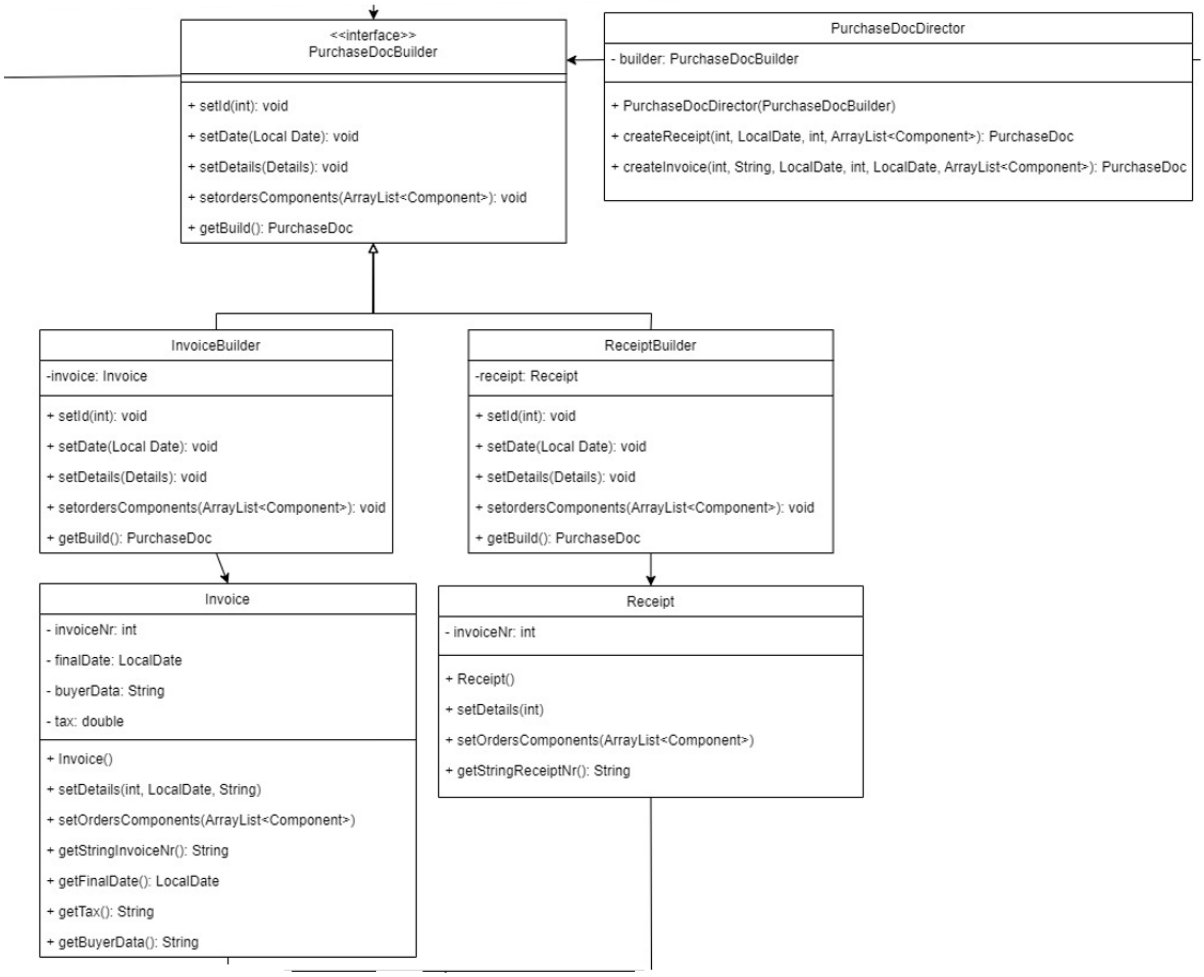
Indywidualne opisy wzorców

- Metoda fabrykująca



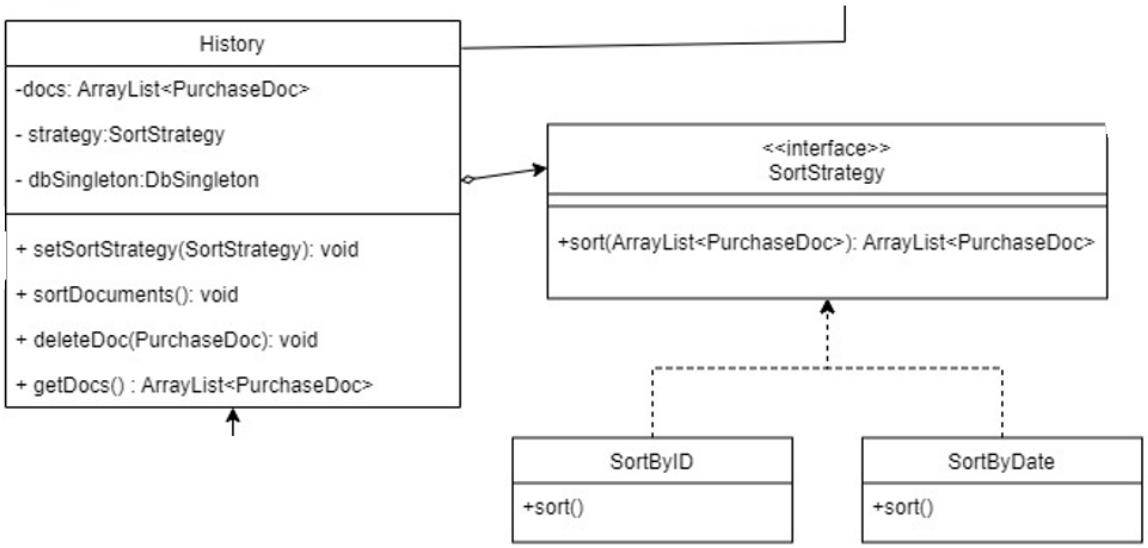
Wzorzec ten został wybrany przez nas do dodawania poszczególnych komponentów do dokumentu zakupu. Metodą fabrykującą w naszym przypadku jest bezparametrowa metoda abstrakcyjna createComponent(). Lokalizacja: pakiet factoryMethod, wektor zmian: możliwość dodania kolejnych komponentówOrderów do rodziny Order oraz implementacje do interfejsu Component

- Builder



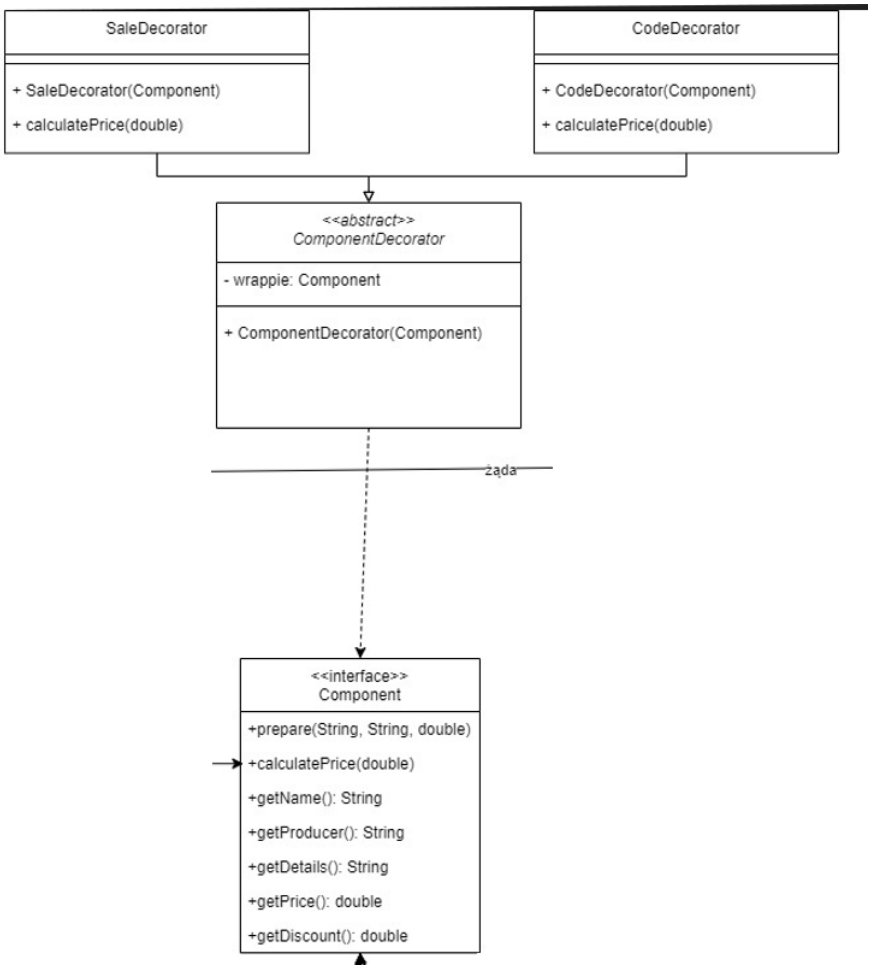
Builder wykorzystamy do tworzenia dokumentu potwierdzającego zakupu danego zamówienia. W zależności od tego czy klient zażyczy sobie fakturę lub paragon taki dokument zostanie utworzony. Klasa, która pełni funkcję director to PurchaseDocDirector. Lokalizacja: pakiet builder, wektor zmian: możliwość dodania nowego builder obok InvoiceBuilder oraz ReceiptBuilder.

- Strategia



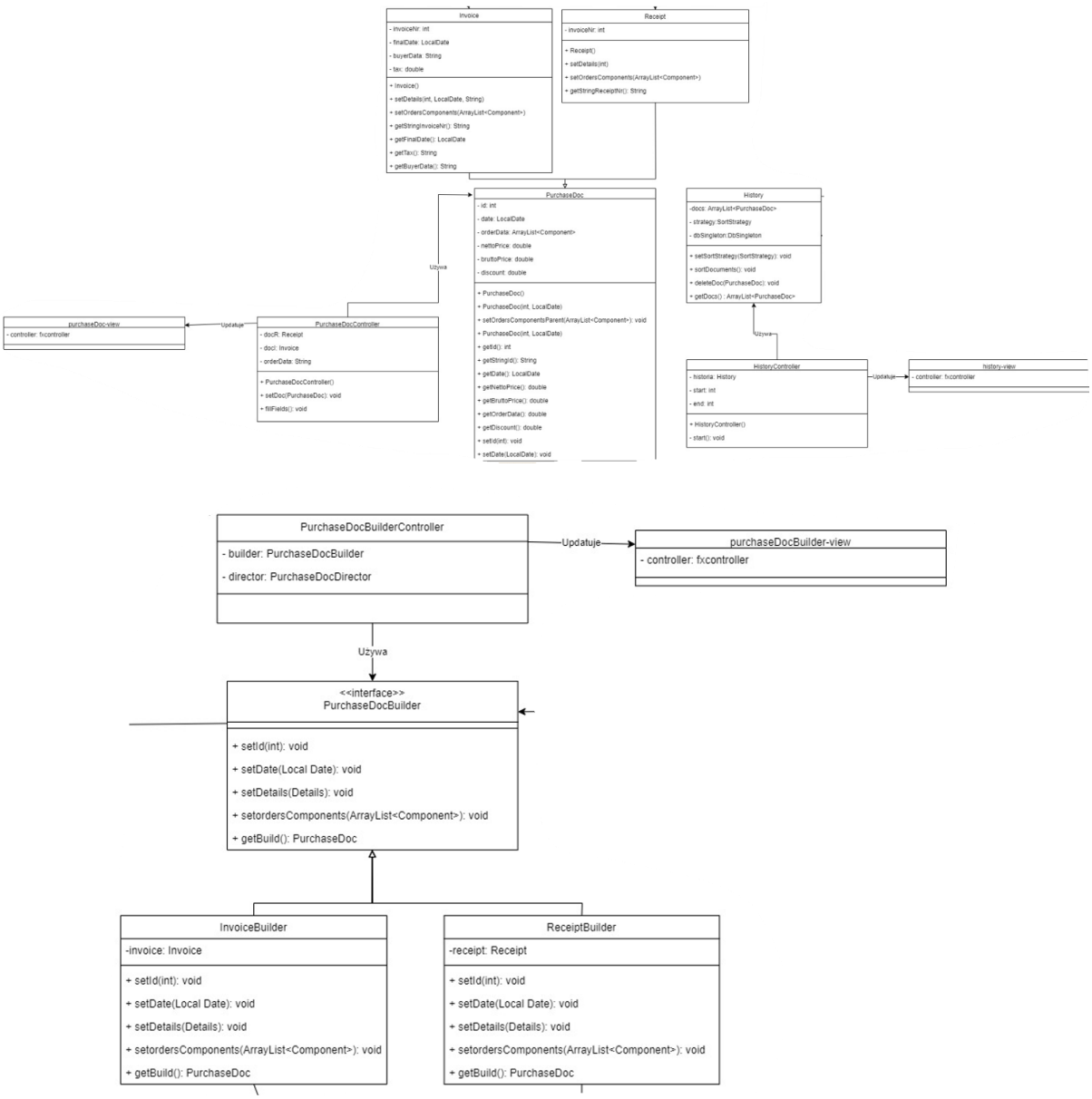
Wzorzec strategii będzie służył nam do odpowiedniego posortowania danych względem danego parametru jak: id dokumentu czy data utworzenia dokumentu. Kontekstem w tym przypadku jest klasa History. Lokalizacja: pakiet strategy, wektor zmian: możliwość dodania nowego sortowania według innego parametru np.po ilosci produktów w zamówieniu, dodanie klasy implementacji SortStrategy

- Dekorator



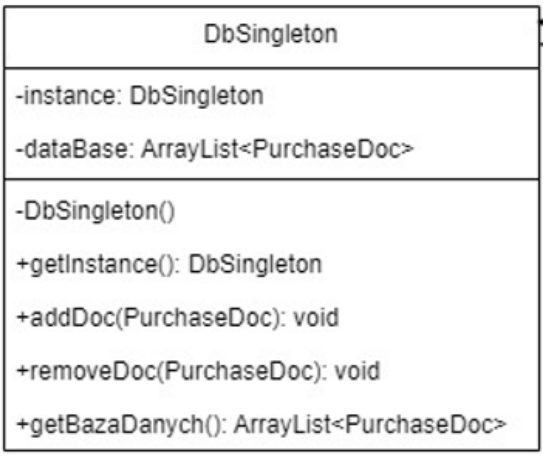
Wzorzec dekoratora będzie służył nam do przydzielania komponentom nowej ceny promocyjnej np. cena wyprzedażowa lub cena uwzględniająca kod rabatowy klienta. Lokalizacja: pakiet decorator, wektor zmian: możliwość dodania innego dekoratora zmieniający cenne danego komponentu np. Zmiana ceny podając procentową zniżkę.

- MVC



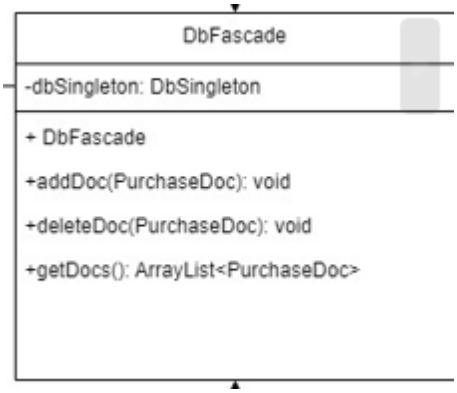
Wzorzec MVC wydziela trzy podstawowe warstwy: Model, View, Controller. Naszymi modelami są np. klasy przedstawiające dokumenty zakupu, czyli Invoice i Receipt, które dziedziczą ze wspólnego rodzica PurchaseDoc oraz klasa History reprezentująca historię dokumentów. Lokalizacja: pakiet models, wszystkie controllery przedstawione na screenie powyzej oraz, pliki fxml w katalogu resources, wektor zmian: możliwość rozbudowy o kolejne widoki np. Dodanie widoku/okienka wyświetlającego się przy usuwaniu dokumentu z bazy z zapytaniem czy na pewno chcemy usunąć wybrany dokument.

- Singleton



Singleton będzie zapewniała wystąpienie w aplikacji tylko jednej instancji dostępu do danych dotyczących dokumentów. Lokalizacja: klasa DbSingleton

- Fasada



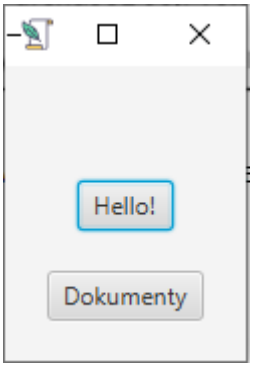
Rolą fasady jest udostępnienie interfejsu do znacznie ułatwionej obsługi bazy danych Singleton. Zamiast wywoływać w całym programie instrukcje operujące bezpośrednio na bazie danych, przygotowana została fasada, która oferuje niezbędne metody do poprawnej komunikacji między aplikacją a bazą danych. Lokalizacja: klasa DbFascade, wektor zmian: możliwość rozszerzenia klasy przy rozszerzaniu mozliwość operowania na bazie danych np. Edycja danych.

Instrukcja instalacji

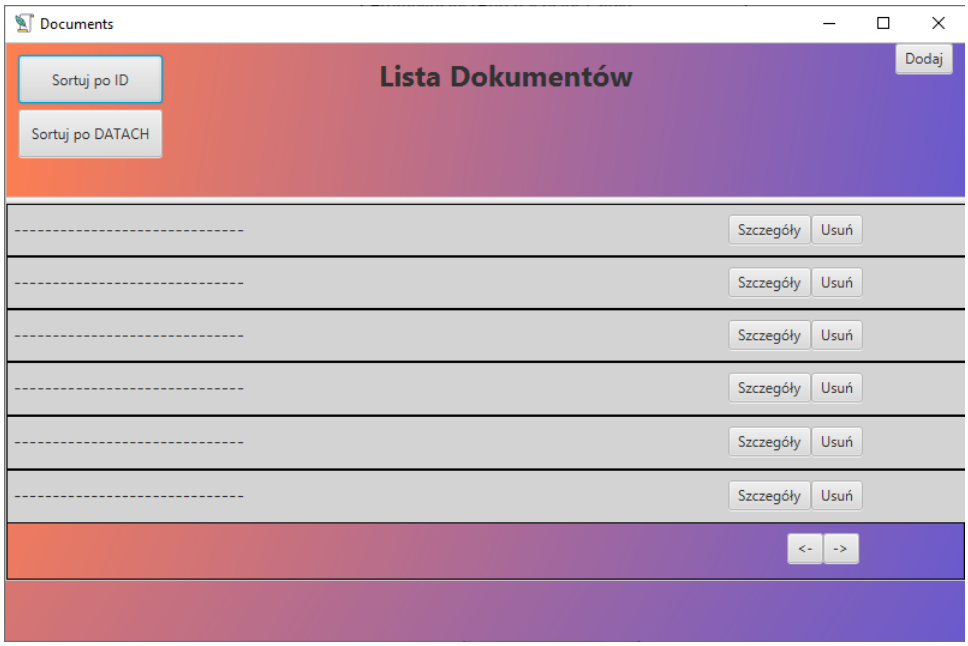
Aby aplikacja poprawnie działała wymagane jest oprogramowanie Java (www.java.com/pl/). Należy pobrać repozytorium, rozpakować je oraz za pomocą wybranego narzędzia(np. intellij idea) otworzyć projekt.

Instrukcja użytkownika

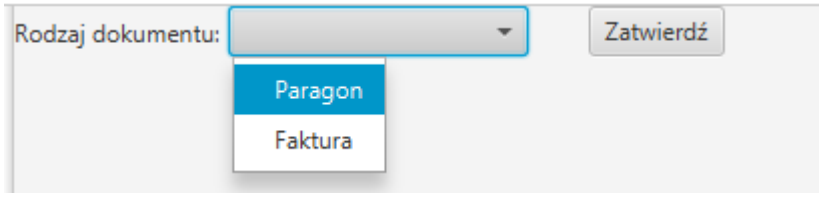
Po uruchomienu aplikacji na ekranie pojawi się małe okienko na którym powinniśmy wygrać przycisk “Dokumenty”

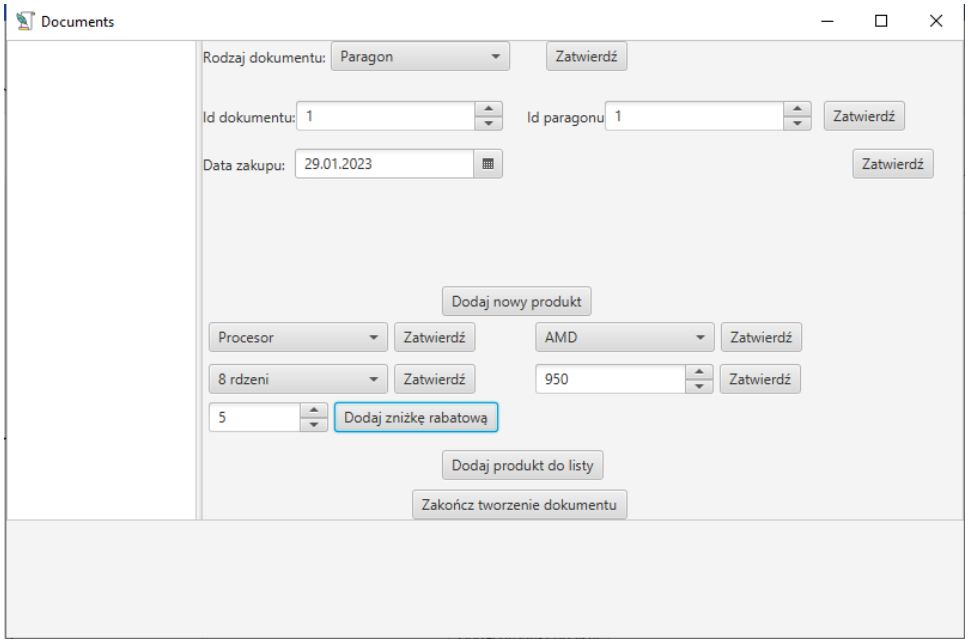


Następnie przekierowuje nas do głównego widoku na którym możemy przeglądać listę dokumentów. Klikając na jeden z przycisków w lewym górnym rogu możemy sortować listę po wybranym parametrze. Jeżeli chce rozpocząć tworzenie dokumentu należy wybrać przycisk “Dodaj” znajdujący się w prawym górnym rogu.

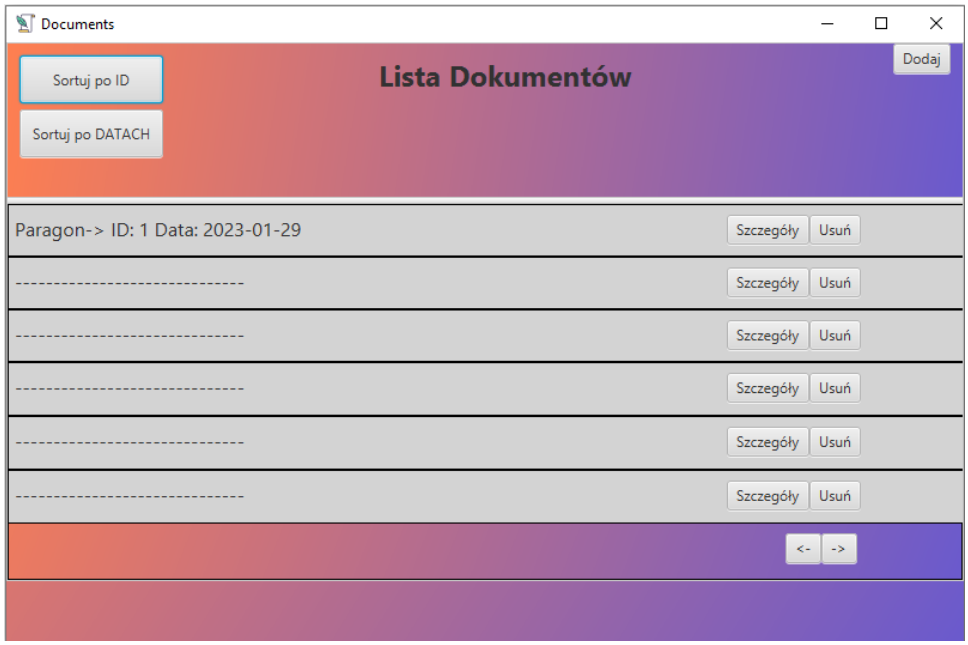


Gdy pojawi się widok tworzenia dokumentu możemy rozpocząć uzupełniać wymagane pola danymi, rozpoczynając od wyboru typu dokumentu.

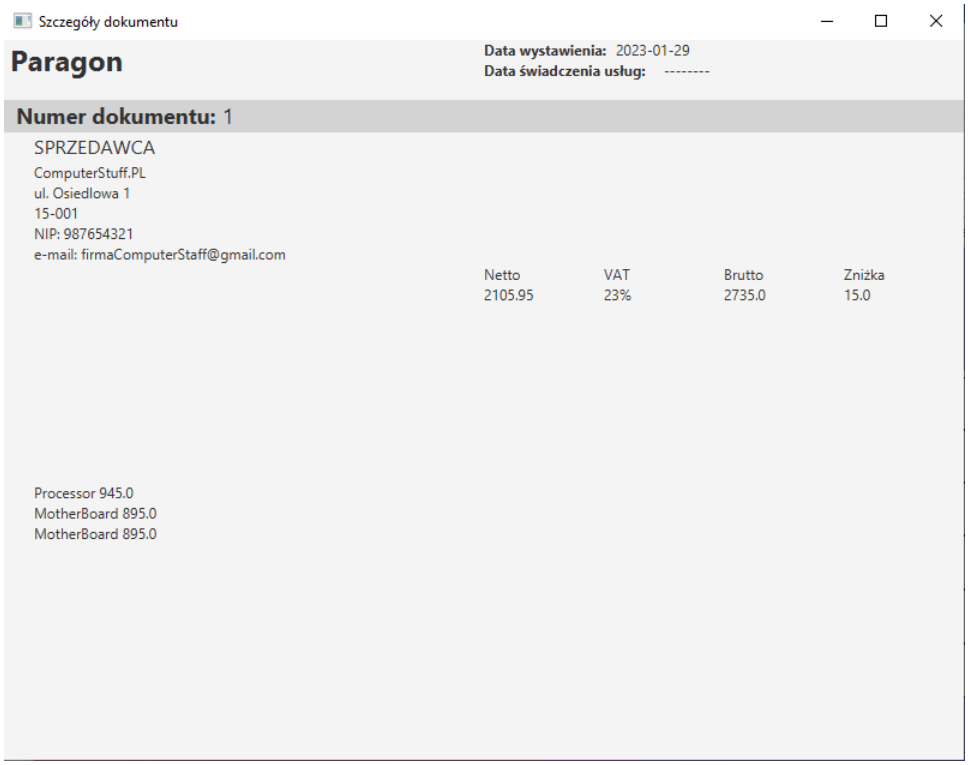


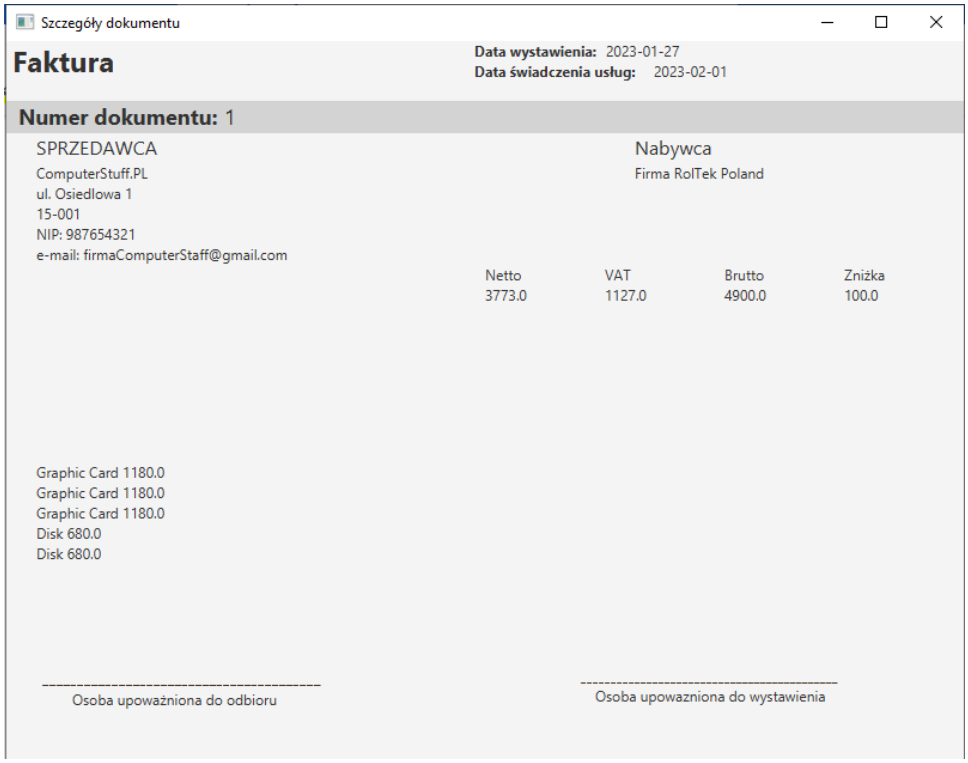


Wybierając opcję “Zakończ tworzenie dokumentu”, wychodzimy z obecnego okna i wracamy do listy dokumentów z już no utworzonym dokumentem.

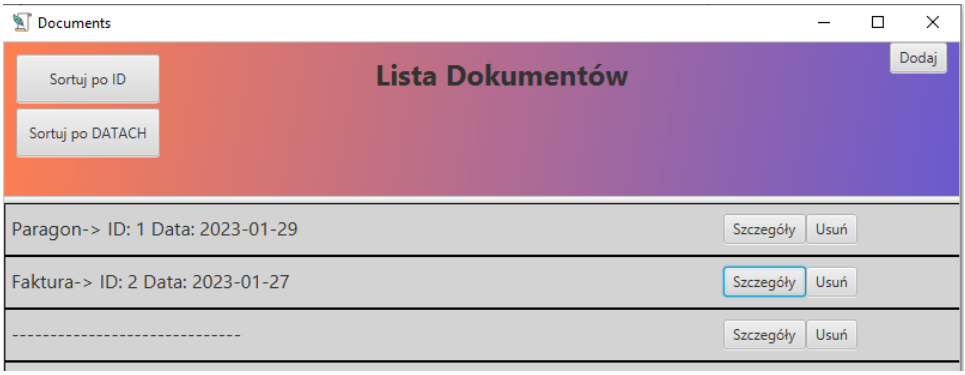


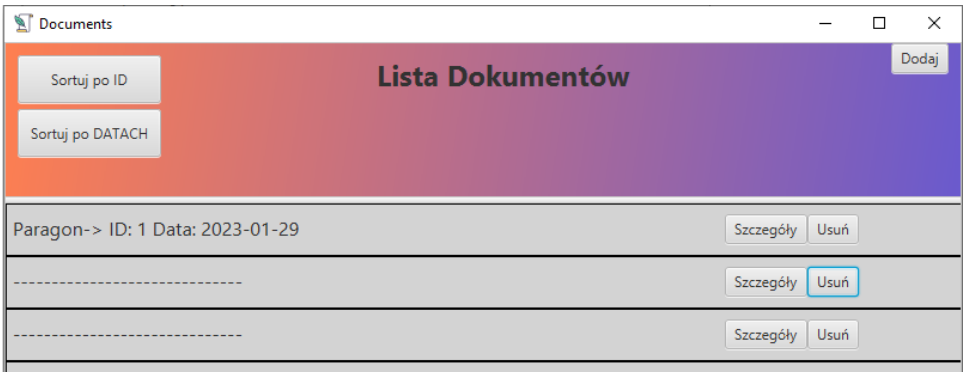
Mając dokumenty w liście mamy możliwość podejrzenia szczegółów dokumentu wybierając przycisk “Szczegóły”.





Istnieje również możliwość usuwania dokumentów z listy wybierając opcję “Usuń”





Przyciski ze strzałkami w prawym dolnym rogu służą do nawigacji pomiędzy stronami listy z dokumentami.

Opis specyficznego rozwiązania w użytej technologii

Zastosowanie technologii JavaFx umożliwiająca tworzenie graficznego interfejsu za pomocą wbudowanego scene buildera lub edytora tekstowego.

Link: https://openjfx.io/