



**WYŻSZA SZKOŁA  
INFORMATYKI i ZARZĄDZANIA**  
z siedzibą w Rzeszowie

## **Projekt**

### **Dokumentacja z przedmiotu ‘Inżyniera oprogramowania’**

**Prowadzący:**

mgr. Tomasz Rajzer

**Student:**

Kamila Tałaj | w64227

**Przedmiot:**

Inżynieria oprogramowania

**Kierunek:**

Informatyka | Technologie internetowe i mobilne

**Rzeszów 2023**

## Spis treści

Dokumentacja zadania "Diagram przypadków użycia" .....	3
Dokumentacja zadania "Diagram klas" .....	4
Dokumentacja zadania "Diagram stanów" .....	6
Dokumentacja zadania "PaymentProcessor Diagram" .....	7
Dokumentacja zadania "Migracje, Encje, API" .....	8
Podsumowanie .....	8

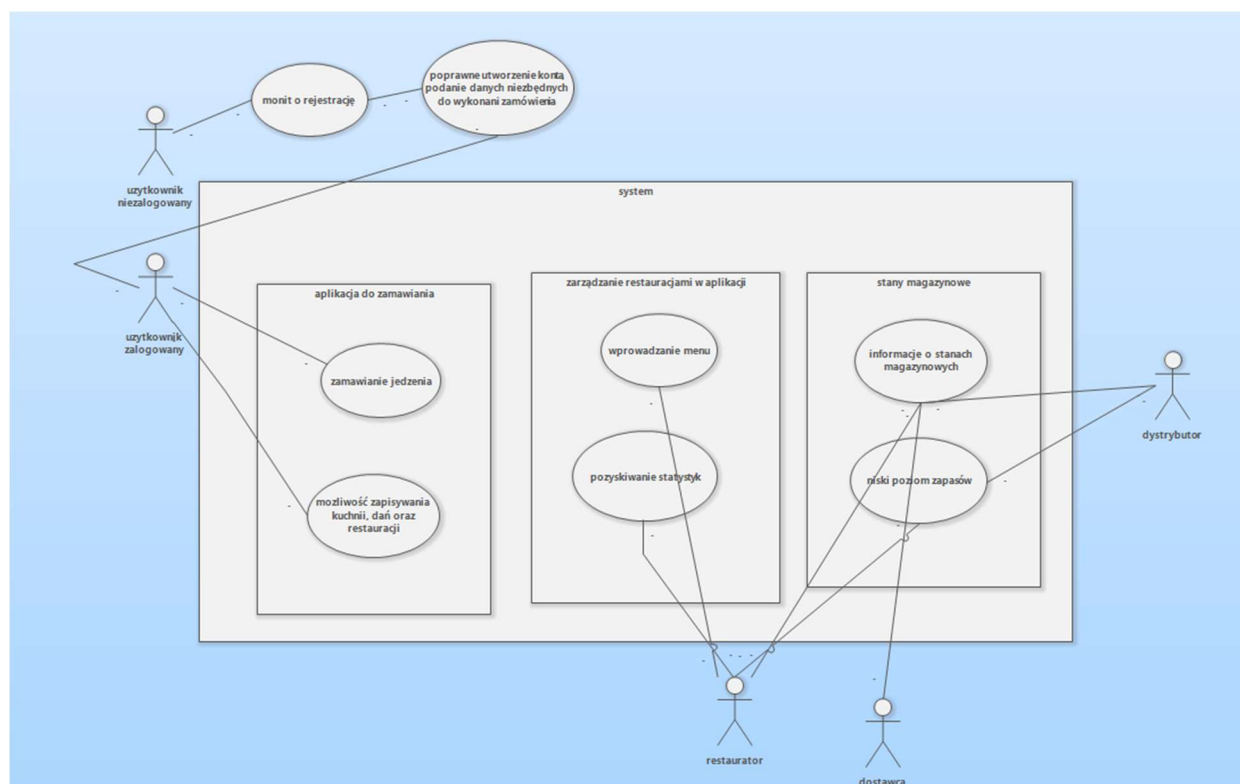
## Dokumentacja zadania "Diagram przypadków użycia"

W ramach tego zadania przeprowadzono analizę aktorów systemu, identyfikując ich potrzeby oraz ograniczenia. Następnie został stworzony diagram przypadków użycia, który przedstawia interakcje między aktorami a systemem.

Celem tego zadania jest opisanie procesu analizy aktorów, identyfikacji ich potrzeb i ograniczeń oraz prezentacja diagramu przypadków użycia.

Zadanie zostało dostosowane do poniższego polecenia:

„Klienci chętnie skorzystają z serwisu umożliwiającego zamawianie jedzenia ale także chcieli by mieć możliwość zapisywania w swoich kontach, ulubionych restauracji oraz ulubionych dań a nawet konkretnych kuchni. Restauratorzy chcą mieć możliwość wprowadzania menu do aplikacji, oraz pozyskiwania statystyk, jakie zarejestrowani użytkownicy najbardziej lubią w danym mieście. Dostawcy chcą posiadać informacje o stanach magazynowych, które regularnie są udostępniane przez dystrybutorów oraz potrzebują wiedzieć, które restauracje mają niskie poziomy zapasów.”



## **Dokumentacja zadania "Diagram klas"**

W tym zadaniu wykonano diagram klas na podstawie wcześniej stworzonego diagramu przypadków użycia. Diagram klas jest graficznym przedstawieniem struktury klas w systemie, relacji między nimi oraz ich atrybutów i metod. Celem tego dokumentu jest opisanie procesu tworzenia diagramu klas oraz przedstawienie uzyskanych wyników.

Przed przystąpieniem do tworzenia diagramu klas, wykonano diagram przypadków użycia, które przedstawiają funkcjonalności systemu oraz interakcje między aktorami a systemem. Diagram użycia stanowi podstawę do identyfikacji klas i relacji między nimi.

Użytkownik ma możliwość:

- Logowania do serwisu,
- Dodania ulubionych potraw i restauracji,
- Podanie danych do złożenia zamówienia.

Restaurator ma możliwość:

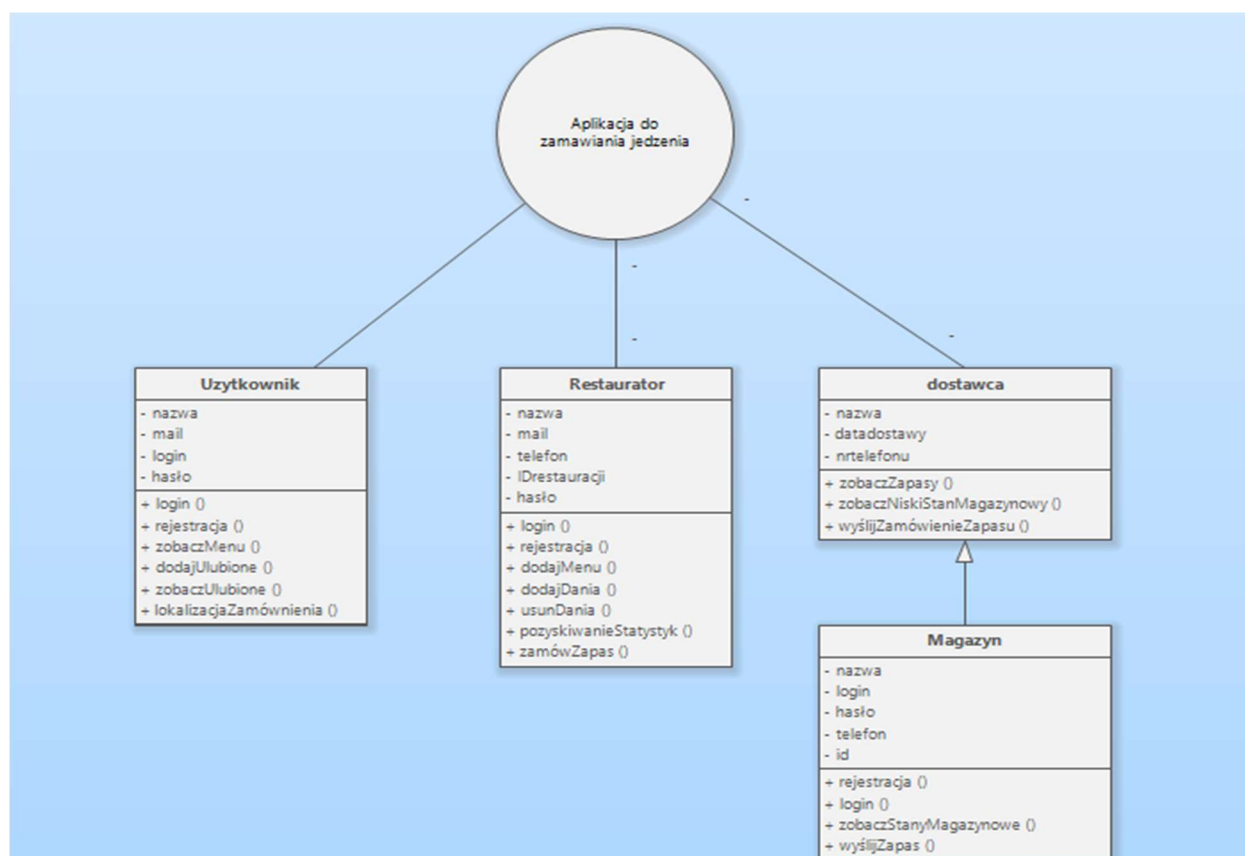
- Logowania do serwisu,
- Dodania / usunięcia potraw do/z menu,
- Pozyskania statystyk z zamówień,
- Zaopatrzenia restauracji.

Dostawca ma możliwość:

- Wglądu w magazyn restauracji,
- Tworzenie wysyłki zaopatrzenia.

Magazyn ma możliwość:

- Wysłania zaopatrzenia do dostawcy po dostaniu monitu o niskich stanach magazynowych.

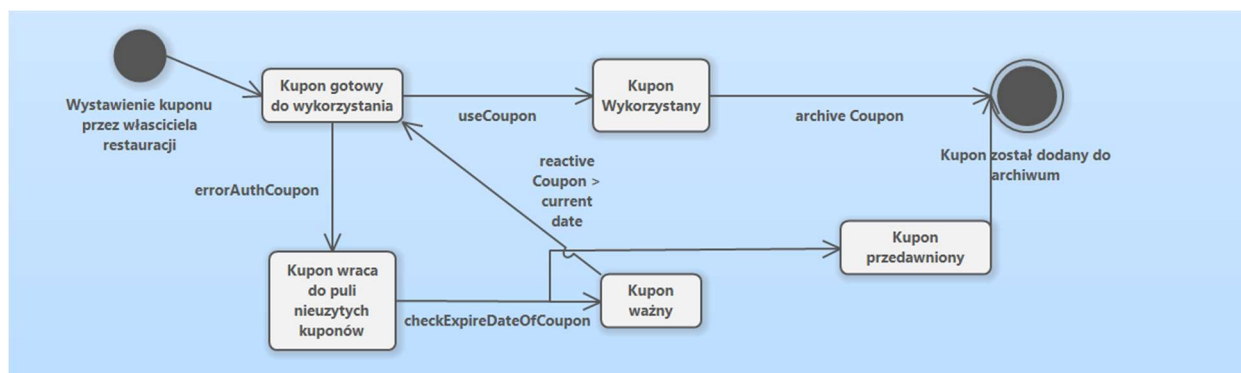


## Dokumentacja zadania "Diagram stanów"

W tym zadaniu utworzono diagram stanów dla nowej funkcjonalności, jaką jest dodanie diagramu stanów użycia kuponów dla branży gastronomicznej do stworzonego w poprzednich diagramach systemu restauracyjnego. Kupon będzie wystawiany przez restauratora i przekazywany użytkownikowi, który będzie mógł go wykorzystać w określonym czasie. Serwis będzie odpowiedzialny za weryfikację ważności kuponu podczas składania zamówienia. Jeśli kupon straci ważność, zostanie przeniesiony do archiwum, gdzie będzie przechowywany dla celów kontroli skarbowej lub ewentualnych zażaleń klienta.

Diagram stanów dla procesu obsługi kuponów w systemie gastronomicznym zawiera poniższe stany:

1. Stan: Gotowy do wykorzystania
  - Opis: Kupon został wystawiony przez restauratora i jeszcze nie został przekazany użytkownikowi.
  - Przejście: Kupon zostaje przekazany użytkownikowi.
2. Stan: Ważny
  - Opis: Kupon został przekazany użytkownikowi i oczekuje na wykorzystanie.
  - Przejścia:
    - Użytkownik wykorzystuje kupon.
    - Kupon traci ważność.
3. Stan: Wraca do puli nieużytych kuponów
  - Opis: Kupon został przekazany użytkownikowi lecz nie został poprawnie użyty/nie został użyty.
  - Przejścia:
    - Kupon wraca do „gotowych do wykorzystania”
    - Kupon terminuje się i przechodzi do archiwum.
4. Stan: Wykorzystany
  - Opis: Użytkownik wykorzystał kupon i otrzymał związane z nim korzyści.
  - Przejście: Archiwum.
5. Stan: Przedawniony
  - Opis: Kupon stracił ważność.
  - Przejście: Kupon zostaje przeniesiony do archiwum.
6. Stan: Archiwum
  - Opis: Kupon jest przechowywany w archiwum dla celów kontroli skarbowej lub ewentualnych zażaleń klienta.
  - Przejście: Brak.



# Dokumentacja zadania "PaymentProcessor Diagram"

Celem tego zadania było utworzenie interfejsu PaymentProcessor, który umożliwi wybór metody płatności za zamówienie. Każda metoda płatności dostarcza informacji o nazwie platformy płatności, spisie opłat za korzystanie oraz listę krajów, do których można przysyłać pieniądze.

Kroki do wykonania:

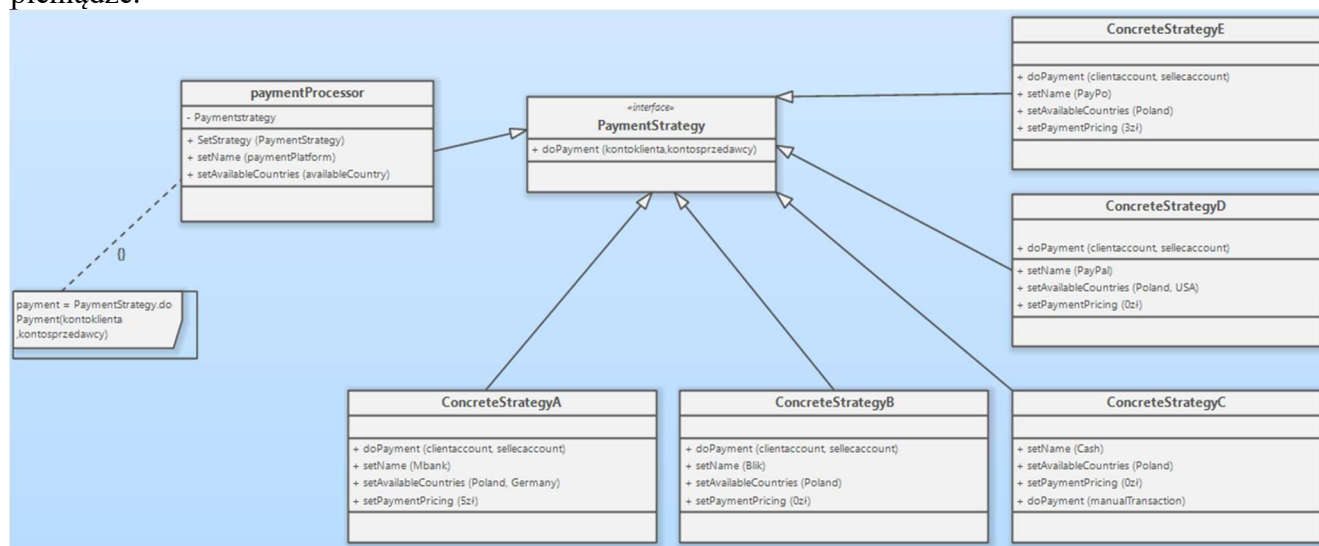
- Utworzenie interfejsu o nazwie PaymentStrategy, który będzie zawierać metody:

setName() - zwraca nazwę platformy płatności,

setPaymentPrices() - zwraca spis opłat za transakcję,

setAvailableCountries() - zwraca listę krajów do których można przysyłać pieniądze.

W wyniku wykonanych czynności utworzono interfejs PaymentProcessor, który umożliwia wybór odpowiedniej metody płatności. Każda metoda płatności dostarcza informacje o nazwie platformy płatności, spisie opłat za korzystanie oraz listę krajów do których można przysyłać pieniądze.



## **Dokumentacja zadania "Migracje, Encje, API"**

Zadanie to polegało na konfiguracji środowiska .NET (służący do konfiguracji przez Swaggera, który umożliwił dostanie się do endpointów) i Angulara (który ostatecznie pozwolił na odczyt endpointów skonfigurowanego API).

Kolejnym krokiem było połączenie środowiska do utworzonej bazy danych w celu wykonania migracji. Wystarczyło wprowadzić dane bazy w odpowiednim pliku, aby komendami terminala przeprowadzić migrację, której wynikiem były zaczytane do bazy tabele.

Ostatnim etapem było uruchomienie nowych klas i metod, które umożliwią dodawanie własnych endpointów do Swaggera. Po wprowadzeniu odpowiedniego kodu i plików, konieczne było wykonanie migracji, która dodaje nową tabelę do przechowywania danych wprowadzanych za pomocą nowo utworzonych endpointów.

## **Podsumowanie**

Zajęcia Inżynierii oprogramowania przyniosły nam wiele różnorodnych tematów, począwszy od testów jednostkowych, aż po tworzenie diagramów, które służą do lepszego zrozumienia działania systemu oraz zawartych w nim elementów.

Link do githuba: <https://github.com/HaxxooR/GPL01>

Na githubie również zamieszczę wszystkie pliki .simp, angular oraz dokumentację.