Wirtualny sprzedawca

Spis treści

[Dziedzina problemowa 2](#_Toc74176388)

[Cel 2](#_Toc74176389)

[Zakres odpowiedzialności systemu 2](#_Toc74176390)

[Użytkownicy systemu 2](#_Toc74176391)

[Wymagania użytkownika 2](#_Toc74176392)

[System przechowuje informacje dotyczące 2](#_Toc74176393)

[Oczekuje się, że system będzie umożliwiał użytkownikom 4](#_Toc74176394)

[System powinien spełniać następujące warunki 4](#_Toc74176395)

[Wymagania funkcjonalne 5](#_Toc74176396)

[Diagram przypadków użycia 5](#_Toc74176397)

[Diagram aktywności 6](#_Toc74176398)

[Diagram kontekstowy 7](#_Toc74176399)

[Analityczny diagram klas 8](#_Toc74176400)

[Wymagania niefunkcjonalne 9](#_Toc74176401)

[Przypadki użycia 10](#_Toc74176402)

[Powielony diagram aktywności przypadków użycia 10](#_Toc74176403)

[Scenariusz przypadku „Utworzenie zamówienia” wraz z przypadkiem „Dodanie karty płatniczej” 11](#_Toc74176404)

[Diagram aktywności 12](#_Toc74176405)

[Diagram stanu 12](#_Toc74176406)

[Diagram interakcji 13](#_Toc74176407)

[Projekt interfejsu dla przypadku użycia 14](#_Toc74176408)

[Skutki analizy dynamicznej 16](#_Toc74176409)

[Poprawiony diagram analityczny 16](#_Toc74176410)

[Omówienie skutków 16](#_Toc74176411)

[Decyzje projektowe 16](#_Toc74176412)

[Diagram projektowy 17](#_Toc74176413)

[Opis przyszłej ewolucji systemu 17](#_Toc74176414)

[Słownik 17](#_Toc74176415)

# Dziedzina problemowa

Z powodu kwarantanny wiele lokali zostało zamkniętych, a pracownicy nie mogą wypełniać swoich obowiązków. Z tego powodu zaistniała potrzeba na umożliwienie lokalom dalszego funkcjonowania spełniając normy bezpieczeństwa, takie jak dostawy bezkontaktowe.

# Cel

Przyśpieszenie transakcji oraz ograniczenie kontaktu z innymi ludźmi poprzez zastąpienie kasjerów stacjonarnych aplikacją w barach na wynos, stacjach benzynowych i wszędzie tam gdzie opłacamy usługę lub kupujemy produkt nie musząc wchodzić w kontakt ze sprzedawcą. Pozwali to utrzymać działanie restauracji, barów oraz drobnych sklepów spożywczych w okresie ścisłej kwarantanny.

# Zakres odpowiedzialności systemu

System powinien pozwalać użytkownikom dokonywać zakupów, tworzyć zamówienia (uwzględniając produkty lub usługi) oraz opłacać je, a pracownikom potwierdzać zamówienia (które mogą zostać zrealizowane).

# Użytkownicy systemu

* Pracownicy lokalu:
  + Operator
  + Realizator
* Edytor
* Użytkownik zarejestrowany
* Gość

# Wymagania użytkownika

### System przechowuje informacje dotyczące

* Klientów
* Pracowników lokalu
* Audytorów
* Lokalów
* Ofert
* Zamówień

|  |  |  |
| --- | --- | --- |
| Klient | Dane personalne | imię |
| nazwisko |
| numer telefonu |
| adres email |
| Dane logowania | login |
| hasło |
| Dane płatnicze | numer karty |
| typ karty |
| CVC |
| data wygaśnięcia |
| Historia zamówień | zamówienia |
| Pracownik lokalu | Dane personalne | imię |
| nazwisko |
| numer telefonu |
| adres email |
| Dane logowania | login |
| hasło |
| Edytor | Dane personalne | imię |
| nazwisko |
| numer telefonu |
| adres email |
| Dane logowania | login |
| hasło |
| Lokal | Dane spółki | nazwa |
| branża |
| Dane lokalu | ulica |
| numer |
| kod pocztowy |
| miejscowość |
| oferta |
| Oferta | Dane oferty | edytor |
| audytor |
| data utworzenia |
| data opublikowania |
| Dane pozycji | nazwa |
| cena |
| Zamówienie | Dane realizacyjne | data utworzenia |
| data realizacji |
| status |
| operator |
| realizator |
| klient |
| lokal |
| Dane oferty | pozycje oferty |
| kwota |

### Oczekuje się, że system będzie umożliwiał użytkownikom

* Przeglądanie oferty – wszyscy
* Rejestrację – gość
* Dodanie karty – klient
* Utworzenie zamówienia – klient
* Zgłaszanie niezgodności zamówienia – klient
* Odczytywanie historycznych zamówień – klient
* Potwierdzenie realizacji zamówienia – pracownik lokalu
* Zarządzanie zamówieniami – operator
* Generowanie raportów sprzedaży – operator
* Zarządzanie pracownikami lokalu – operator
* Zarządzanie ofertą – edytor

### System powinien spełniać następujące warunki

* Dla serwera:
  + Obsługuje jednocześnie minimum 100 zapytań
  + Tworzy regularnie kopie zapasowe
  + Dane wrażliwe są szyfrowane
  + Podczas awarii powinien być naprawiony w ciągu 24h
* Dla klienta mobilnego:
  + Logo firmy w lewym górnym rogu
  + Możliwość wyboru szaty graficznej
  + Regularne aktualizacje bezpieczeństwa
  + Aktualizacje wdrażane stopniowo
  + GUI w wielu językach

# Wymagania funkcjonalne

### Diagram przypadków użycia

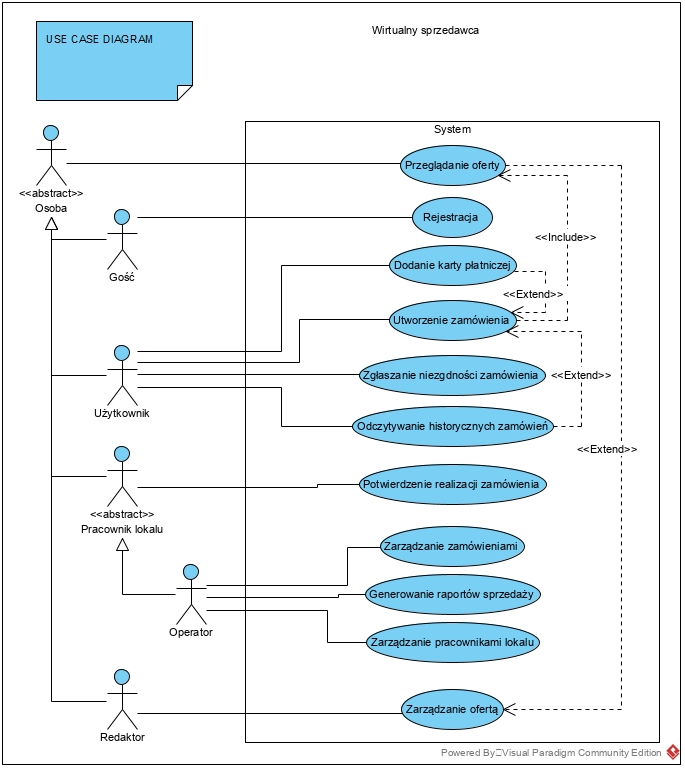
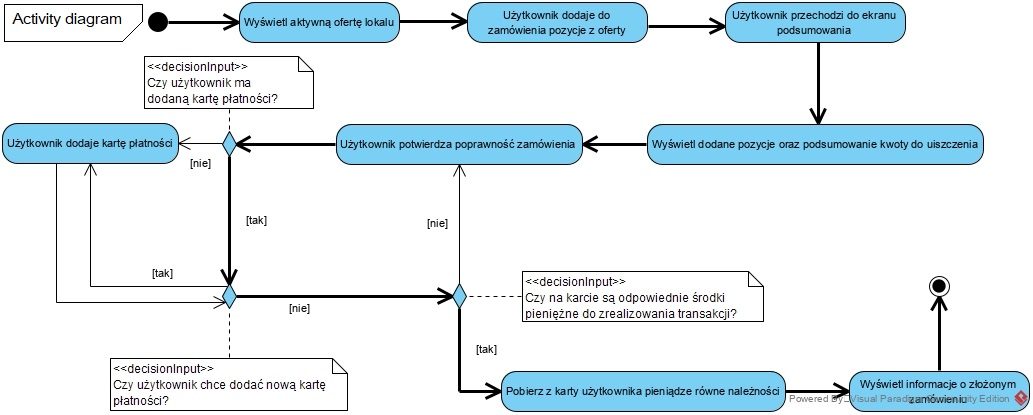
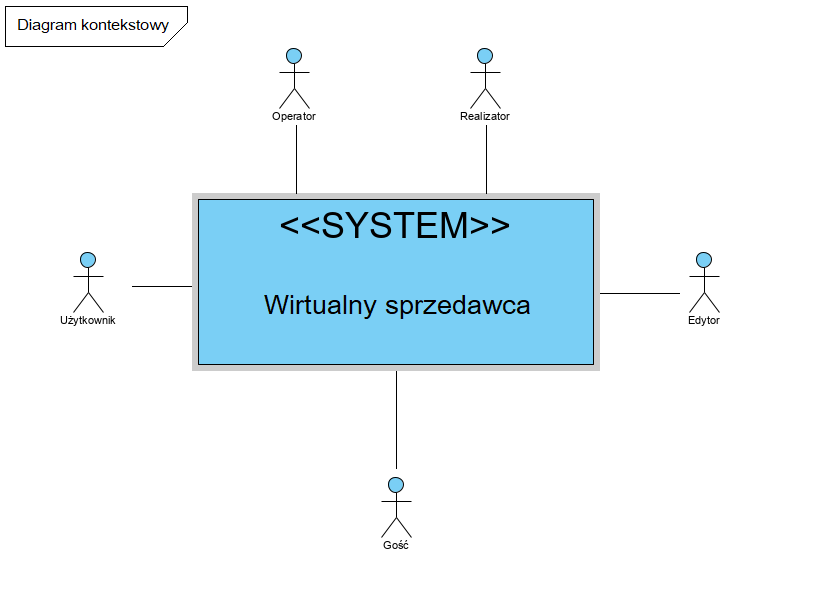


Diagram aktywności

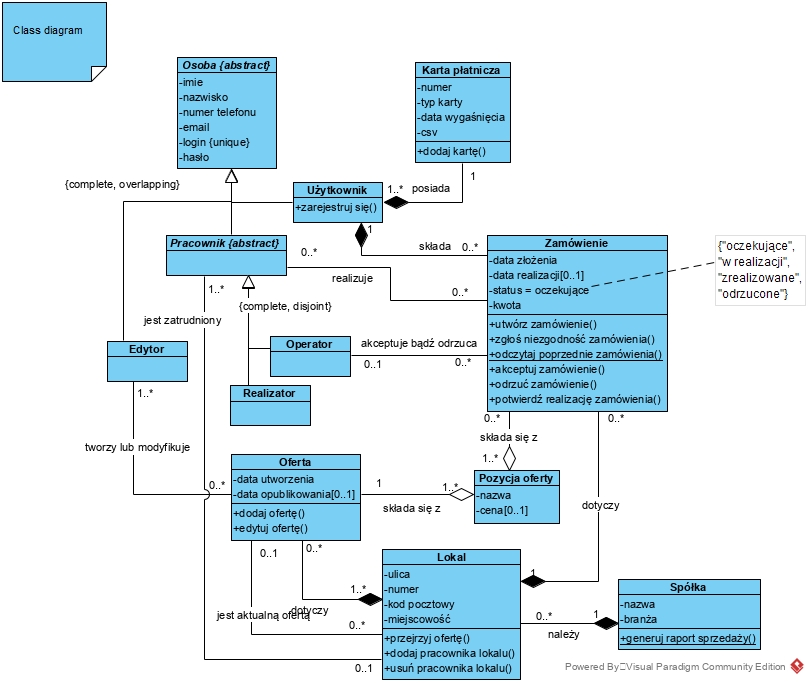


|  |  |
| --- | --- |
| Nazwa przypadku użycia | Utworzenie zamówienia |
| Warunek początkowy | Użytkownik jest zalogowany oraz posiada co najmniej jedną kartę płatniczą  Użytkownik znajduje się w zasięgu lokalu |
| Główny przepływ zdarzeń | 1. Przypadek użycia zostaje uruchomiony przez użytkownika 2. System wyświetla aktywną ofertę danego lokalu 3. Użytkownik dodaje do swojego zamówienia pozycje z oferty 4. Użytkownik kończy dodawanie pozycji poprzez przejście do ekranu podsumowania 5. System wyświetla aktualny koszyk użytkownika w formie podsumowania, wraz z sumą kwoty do uiszczenia 6. Użytkownik potwierdza wprowadzone pozycje 7. System pobiera z karty użytkownika pieniądze za zamówienie 8. System informuje o złożeniu zamówienia |
| Alternatywny przepływ zdarzeń | 6a. Użytkownik dodaje nową kartę płatności  6b. Użytkownik wybiera kartę z jakiej chce dokonać opłaty za zamówienie  7a. Na karcie nie ma dostępnych środków, system wyświetla odpowiednią informację i cofa przepływ zdarzeń do punktu 5 |
| Zakończenie | W dowolnej chwili |
| Warunek końcowy | W przypadku utworzenia zamówienia, do systemu wprowadzone zostaje zgłoszenie wraz z zawieranymi pozycjami |

### Diagram kontekstowy



# Analityczny diagram klas

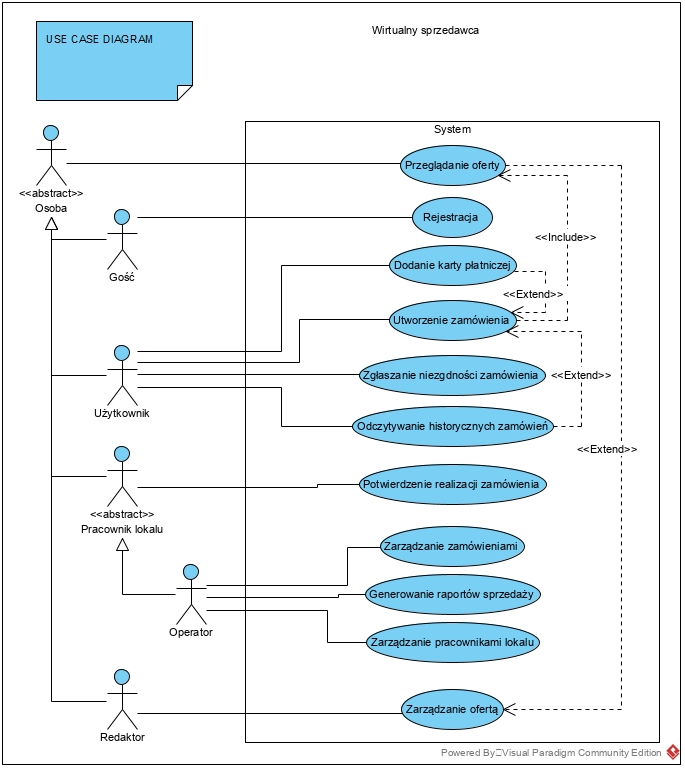


# Wymagania niefunkcjonalne

* Dla serwera:
  + Obsługuje jednocześnie minimum 100 zapytań:
    - Baza danych oparta na MSSQL a całość na dwóch oddzielnych maszynach z Windows Server 2016 lub nowszym. Maszyny o minimalnej konfiguracji 8 GB RAM i 16 rdzeni o taktowaniu 2 GHz
  + Tworzy regularnie kopie zapasowe:
    - Co tydzień pełna kopia
    - Codziennie kopia różnicowa
  + Dane wrażliwe są szyfrowane:
    - Algorytm SHA256
  + Podczas awarii powinien być naprawiony w ciągu 24h
    - Spełniana jest norma niezawodności PN-IEC 60300
* Dla klienta mobilnego:
  + Logo firmy w lewym górnym rogu:
    - W formie przycisku menu
  + Możliwość wyboru szaty graficznej:
    - Dostępne dwa motywy, jasny oraz ciemny
  + Regularne aktualizacje bezpieczeństwa:
    - Aplikacja spełnia normy bezpieczeństwa ISO/IEC 27001
  + Aktualizacje wdrażane stopniowo:
    - Aktualizacja dziennie pokrywa maksymalnie 10% wszystkich użytkowników
  + GUI w wielu językach:
    - Polski
    - Angielski
    - Rosyjski

# Przypadki użycia

## Powielony diagram aktywności przypadków użycia



## Scenariusz przypadku „Utworzenie zamówienia” wraz z przypadkiem „Dodanie karty płatniczej”

|  |  |
| --- | --- |
| Nazwa przypadku użycia | Utworzenie zamówienia |
| Warunek początkowy | Użytkownik jest zalogowany oraz posiada co najmniej jedną kartę płatniczą  Użytkownik znajduje się w zasięgu lokalu |
| Główny przepływ zdarzeń | 1. Przypadek użycia zostaje uruchomiony przez użytkownika 2. System wyświetla aktywną ofertę danego lokalu 3. Użytkownik dodaje do swojego zamówienia pozycje z oferty 4. Użytkownik kończy dodawanie pozycji poprzez przejście do ekranu podsumowania 5. System wyświetla aktualny koszyk użytkownika w formie podsumowania, wraz z sumą kwoty do uiszczenia 6. Użytkownik potwierdza wprowadzone pozycje 7. System pobiera z karty użytkownika pieniądze za zamówienie 8. System informuje o złożeniu zamówienia |
| Alternatywny przepływ zdarzeń | 6a. Użytkownik dodaje nową kartę płatności  6b. Użytkownik wybiera kartę z jakiej chce dokonać opłaty za zamówienie  7a. Na karcie nie ma dostępnych środków, system wyświetla odpowiednią informację i cofa przepływ zdarzeń do punktu 5 |
| Zakończenie | W dowolnej chwili |
| Warunek końcowy | W przypadku utworzenia zamówienia, do systemu wprowadzone zostaje zgłoszenie wraz z zawieranymi pozycjami |

Scenariusz rozpoczyna się w momencie kiedy zalogowany użytkownik chce dokonać zakupu z lokalu.

Użytkownik przeglądając ofertę lokalu wybiera interesujące go pozycje. Spowoduje to dodanie tych pozycji do koszyka.

Gdy użytkownik zdecyduje o chęci sfinalizowania zakupu, przechodzi do podsumowania, na którym widzi wybrane przez siebie pozycje wraz z kwotą łączną, a system sprawdza poprawność zamówienia.

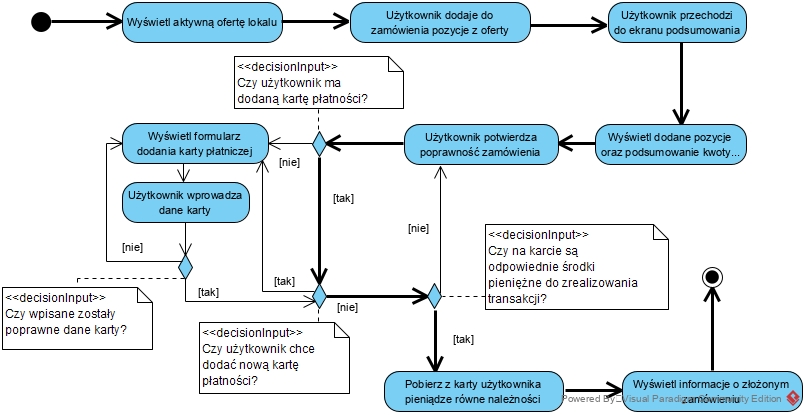
Sprawdzanie poprawności zamówienia polega na upewnieniu się że lokal wciąż posiada w ofercie odpowiednie produkty, w celu wyeliminowania kupna wycofanej w międzyczasie pozycji.

Potwierdzając finalizowanie zakupu użytkownik przechodzi do ekranu płatności, na którym wybrać musi kartę z której chce opłacić zakup.

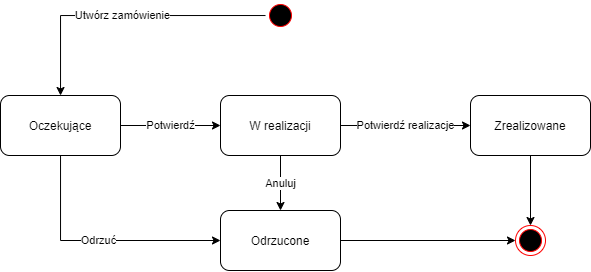
Jeśli użytkownik nie ma dodanej karty płatniczej musi ją dodać przed przejściem do następnego kroku, w innym przypadku działanie to jest opcjonalne. Dodanie karty płatności polega na wyświetleniu formularza na numer karty, datę ważności oraz kod CVC. Przed zapisaniem karty w bazie system weryfikuje poprawność wpisanych informacji, następnie blokuje na krótką chwilę 10 groszy na karcie w celu przetestowania działania karty.

Po wybraniu odpowiedniej opcji płatności system ponownie sprawdza poprawność zamówienia, zapisuje jego dane w systemie, a następnie obciąża kartę klienta zadaną kwotą. Po pobraniu płatności aplikacja wyświetla potwierdzenie i prognozowany czas wykonania usługi.

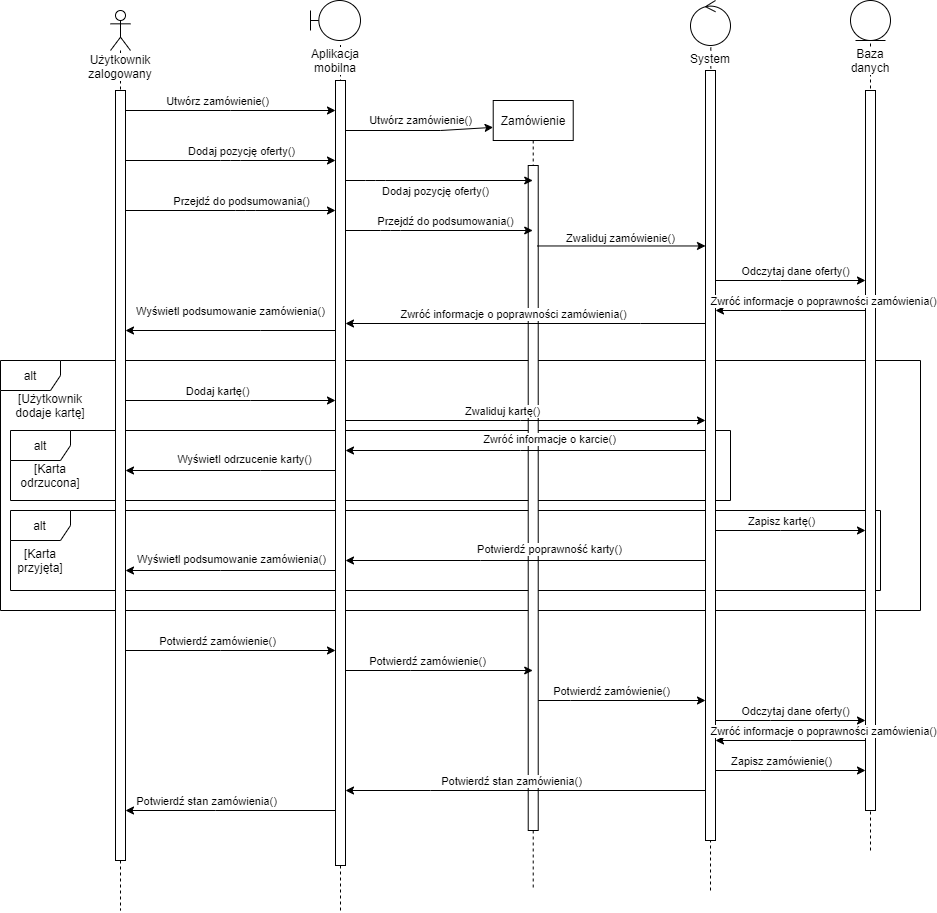
## Diagram aktywności



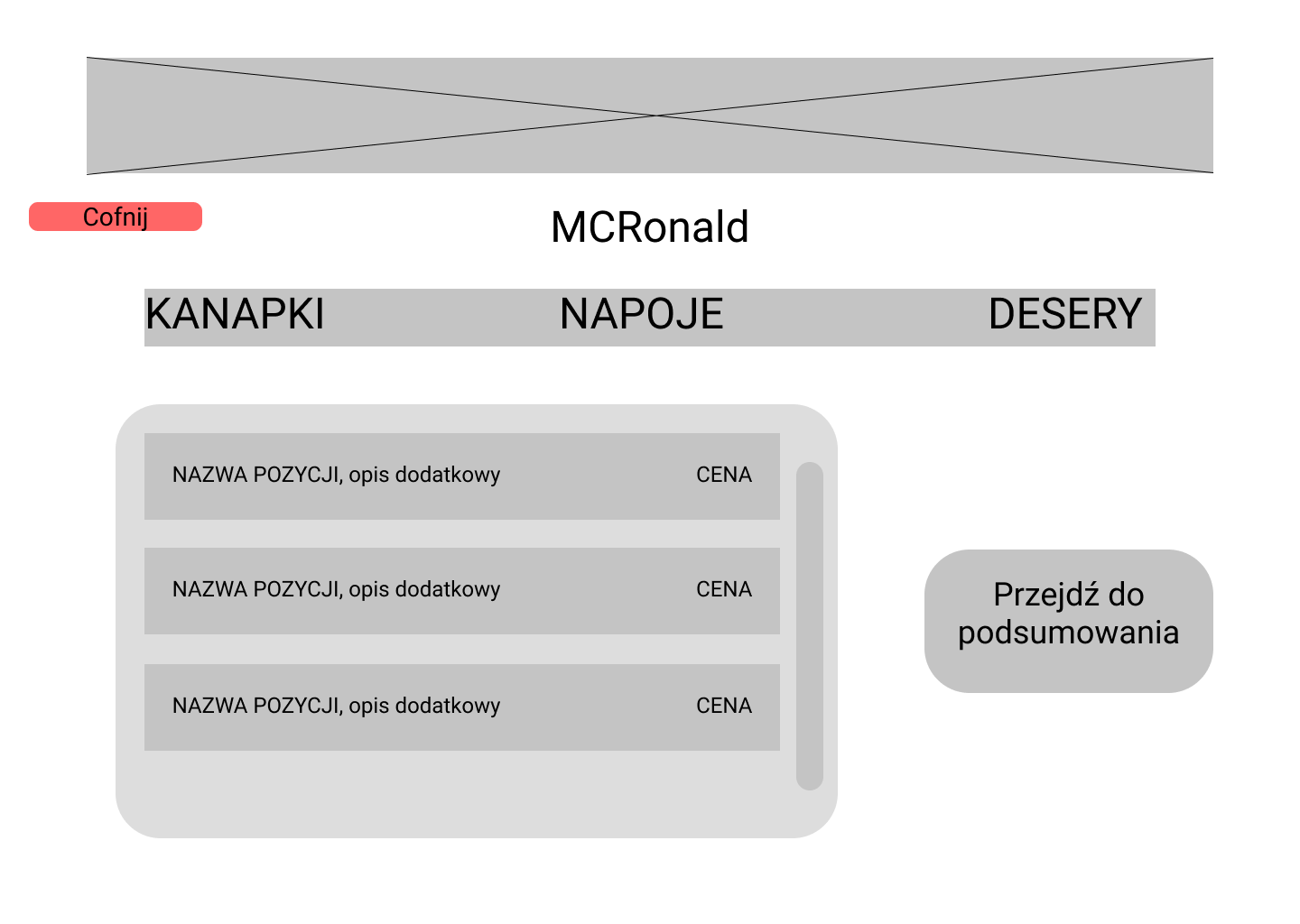
## Diagram stanu

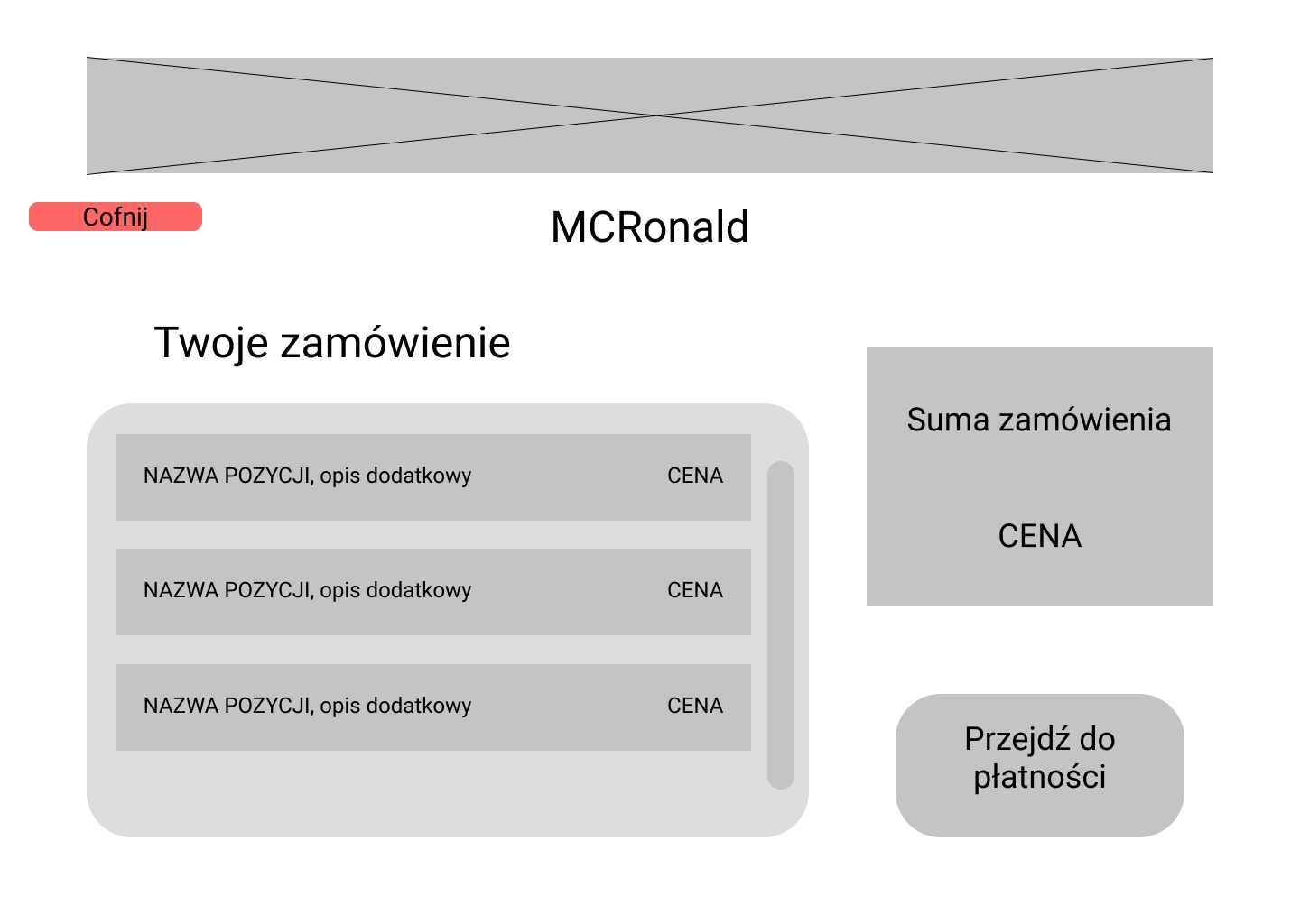


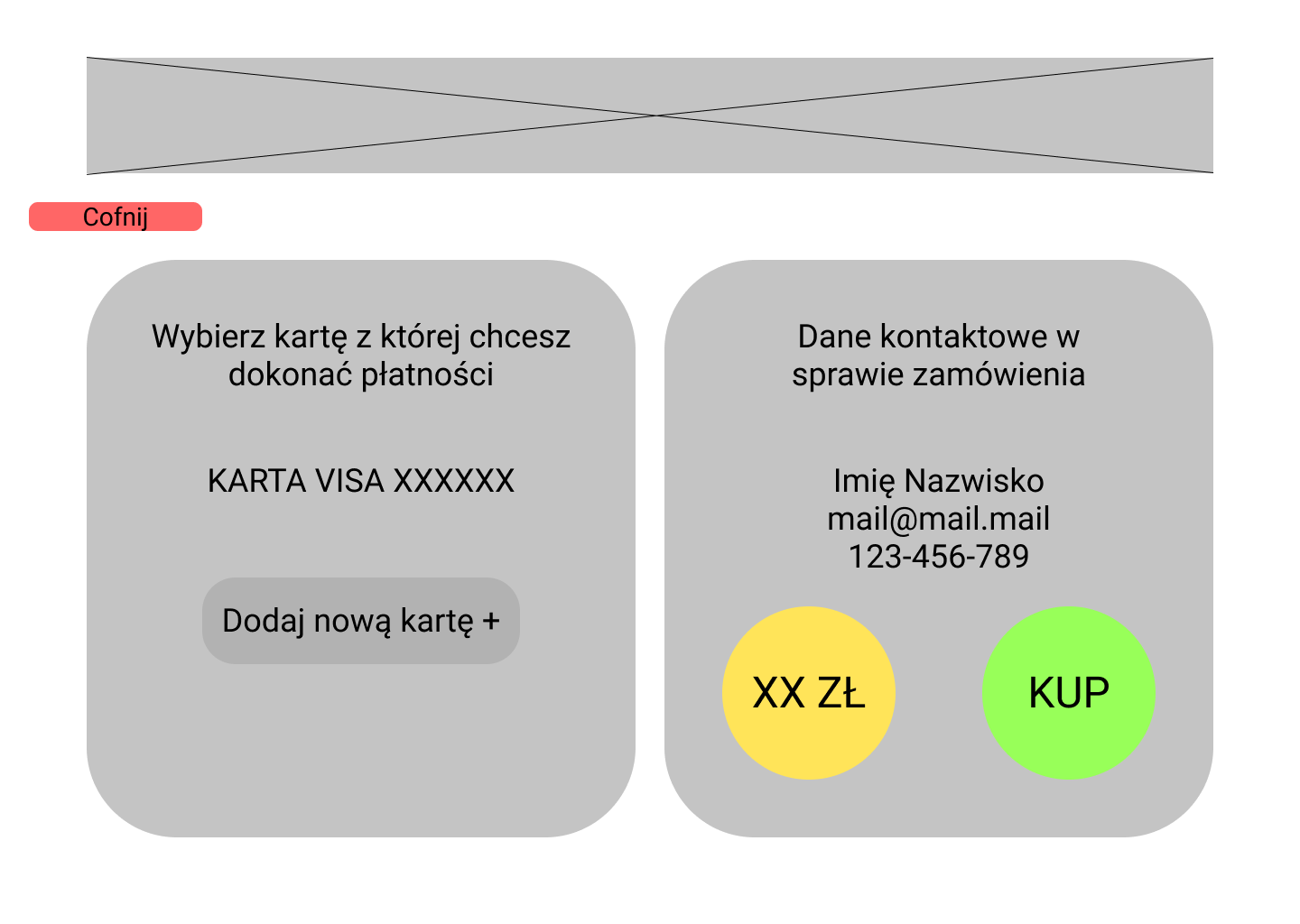
## Diagram interakcji



## Projekt interfejsu dla przypadku użycia

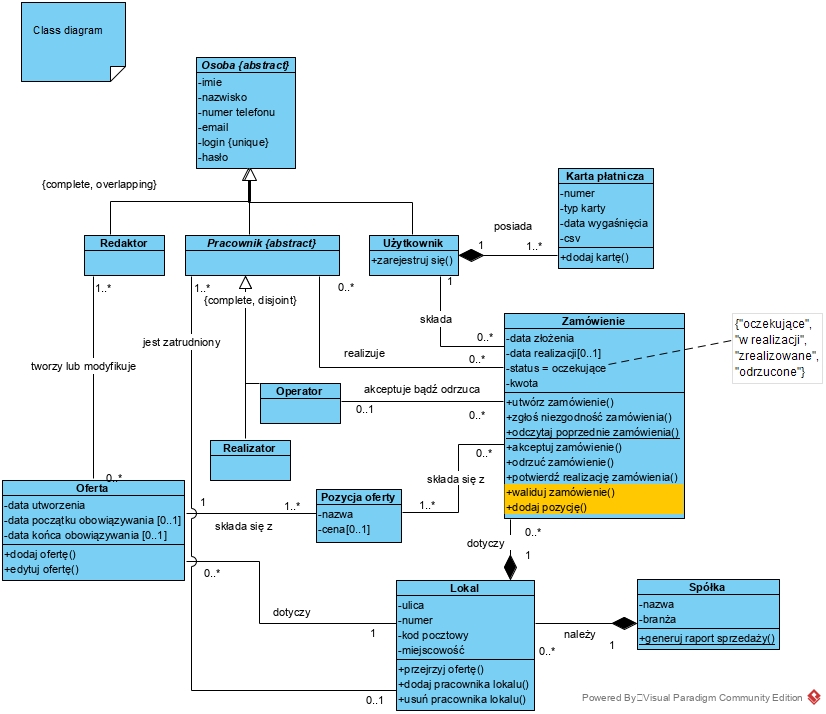






# Skutki analizy dynamicznej

## Poprawiony diagram analityczny



## Omówienie skutków

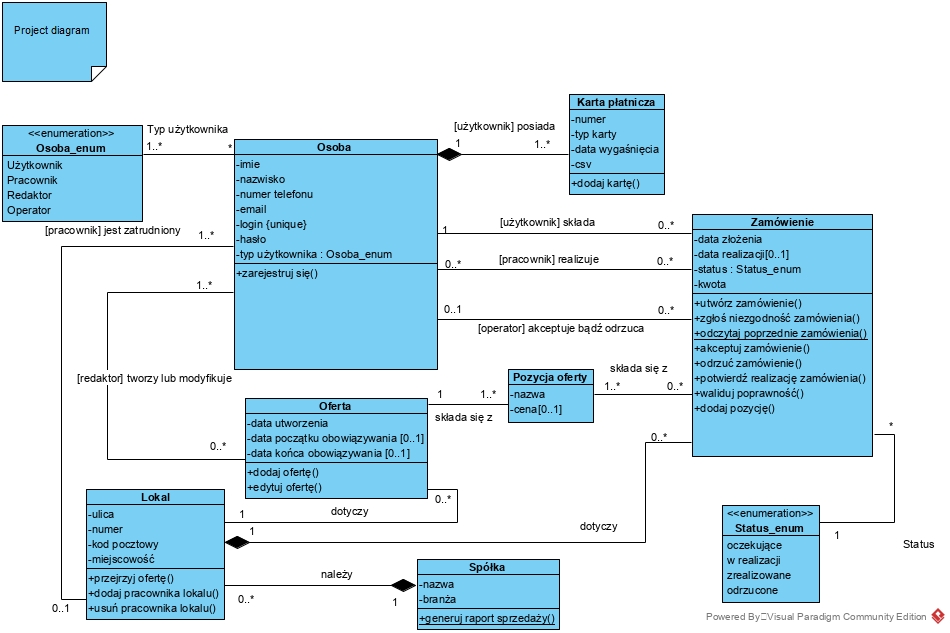
Po dokładnym przeanalizowaniu scenariusza okazało się, że w projekcie brakuje bardzo ważnej metody w klasie Zamówienie – walidowania. Metoda jest bardzo ważna, uruchamiana jest aż dwa razy zanim zamówienie zostanie zapisane w bazie danych.

Zabrakło również metody dodawania pozycji oferty do zamówienia, co znacząco ułatwi implementację projektu.

# Decyzje projektowe

Ze względu na ograniczenia narzucone przez język programowania i sposób przechowywania informacji w bazach relacyjnych, klasy projektu odpowiadające za użytkowników systemu musiały zostać spłaszczone do jednej klasy – Osoba. Overlapping wymusił skorzystanie ze sposobu implementacji opartym na enumach, czyli każdy obiekt klasy Osoba przypisaną ma listę z wartościami typu enum określającymi typ użytkownika. Ze względu na dodatkowe dziedziczenie disjoint pomiędzy operatorem a realizatorem, zostanie zastosowane ograniczenie, polegające na zablokowaniu możliwości dodania dwóch roli z tego dziedziczenia do jednej osoby. Status zamówienia również zostanie zaimplementowany jako typ enum.

# Diagram projektowy



# Opis przyszłej ewolucji systemu

Jeśli projekt zostanie zaakceptowany przez społeczność, planowany jest rozwój w stronę:

* Większa ilość języków
* Możliwość wybrania między motywami ciemnym i jasnym
* Dodanie typu użytkownika administrator – zajmowałby się administrowaniem spółki, jej lokalami i pracownikami wewnątrz nich

# Słownik

Operator: Pracownik o podwyższonych uprawnieniach, kierownik lokalu lub oddelegowany przez niego pracownik. Jego zadaniem jest akceptowanie bądź odrzucanie zamówień wysyłanych przez klientów, oraz wybieranie aktywnej oferty.

Oferta: Spis pozycji składających się na ofertę danego lokalu, ofert dla jednego lokalu może istnieć kilka ale tylko jedna jest aktywna.

Redaktor (Edytor w starej części dokumentacji): Użytkownik którego zadaniem jest utworzenie oraz modyfikacja ofert lokali.