# **CookBook** – książka kucharska

*Nikodem Kwaśniak i Łukasz Kosmaty*

## Spis treści

## Opis projektu:

Aplikacja frontendowo-backendowa wykorzystująca zewnętrzne api, z którym komunikuje się poprzez zapytania http.
Projekt zostanie podzielony na 3 osobne podsystemy:

- **front-end** - będzie odpowiedzialna za interakcję z użytkownikiem oraz za poprawne wyświetlanie danych. Część ta korzystać będzie z responsywnego podejścia w celu zapewnienia poprawnego wyświetlania treści na urządzeniach desktopowych i mobilnych.
- **serwer** - jego zadaniem będzie wzbogacenie zewnętrznego api o dodatkowe funkcjonalności oraz przekierowanie niektórych zapytań do wcześniej wymienionego api.
- **generator kodu klienta** - generator kodu typescriptowego z pliku json wystawianego przez część serwerową. Posiadać on będzie prosty interfejs graficzny.

## Oczekiwane działanie:

*Frontend***:**
Po wpisaniu w wyszukiwarce określonych składników aplikacja zwraca listę przepisów, gdzie można wykorzystać owe produkty, a także listę przydatnych do tego akcesoriów kuchennych. Dodatkowo pod każdym z przepisów użytkownik może zostawić komentarz po wcześniejszym zarejestrowaniu lub zalogowaniu się.
*Generator kodu klienta:*
Po podaniu odpowiedniego pliku json ma zostać wygenerowany kod typscriptowy umożliwiający połączenie się z serwerem oraz wykonanie wymaganych przez frontend zapytań http.
*Serwer:*
Powinien udostępniać proste w obsłudze rest api umożliwiające interakcję z bazą danych oraz wyszukiwanie przepisów.

## Funkcjonalności*:*

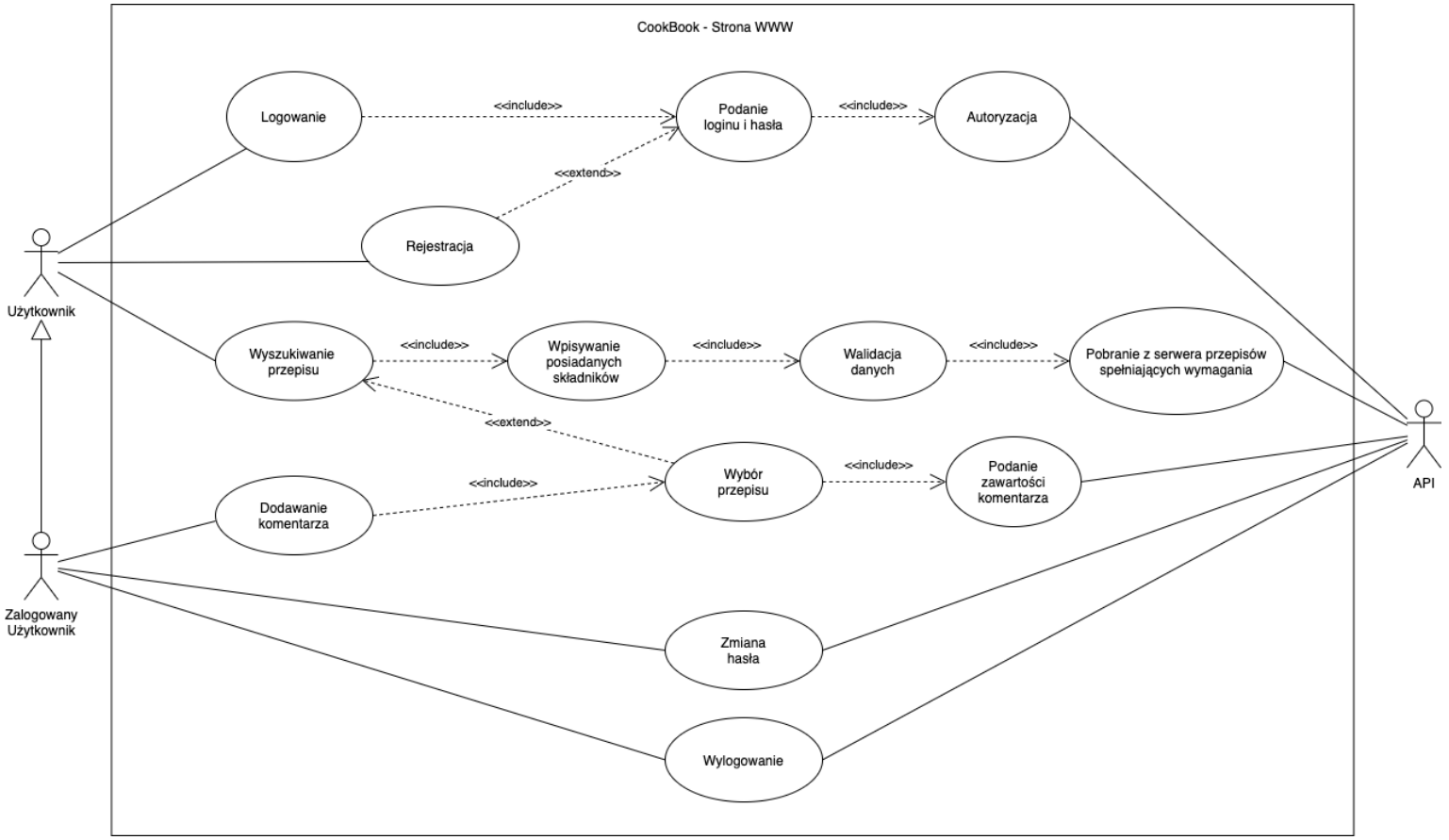
- rozdzielanie listy składników za pomocą znaków specjalnych (przecinek) lub białych znaków (spacja, enter)
- rejestracja użytkownika z uwzględnieniem silności wprowadzanego przez niego hasła
- komentowanie przepisu po zalogowaniu się
- możliwość ustawienia nazwy, zdjęcia profilowego przy rejestracji użytkownika
- możliwość zmiany nazwy, zdjęcia profilowego po rejestracji użytkownika
- możliwość usuwania z wyszukiwarki danych produktów przy pomocy specjalnego elementu UI
- menu przepisu z rozwijanym opisem, listą składników i akcesoriów potrzebnych do przygotowania dania
- logowanie
- wylogowanie się usera z aplikacji
- pasek nawigacyjny z uwzględnieniem paneli do opisu aplikacji oraz kontaktu
- generowanie kody TS
- generator powinien obsługiwać 70% standardu open api - http://spec.openapis.org/oas/v3.0.1 (wybranej wersji)
- serwer powinien cachować niektóre odpowiedzi
- generator powinien być rozszerzalny (w przyszłości) o nowe wersję standardu api

## Ustalenia techniczne:

| | Język programowania | Frameworki/biblioteki |
|---|---|---|
| Front-end | JavaScript, SASS, TypeScript | React, Redux, React-Router |
| Serwer | C# | ASP.CORE, EntityFramework, Swagger, AutoMapper, FluentValidator |
| Generator kodu klienta | C# | T4, Gui.cs |

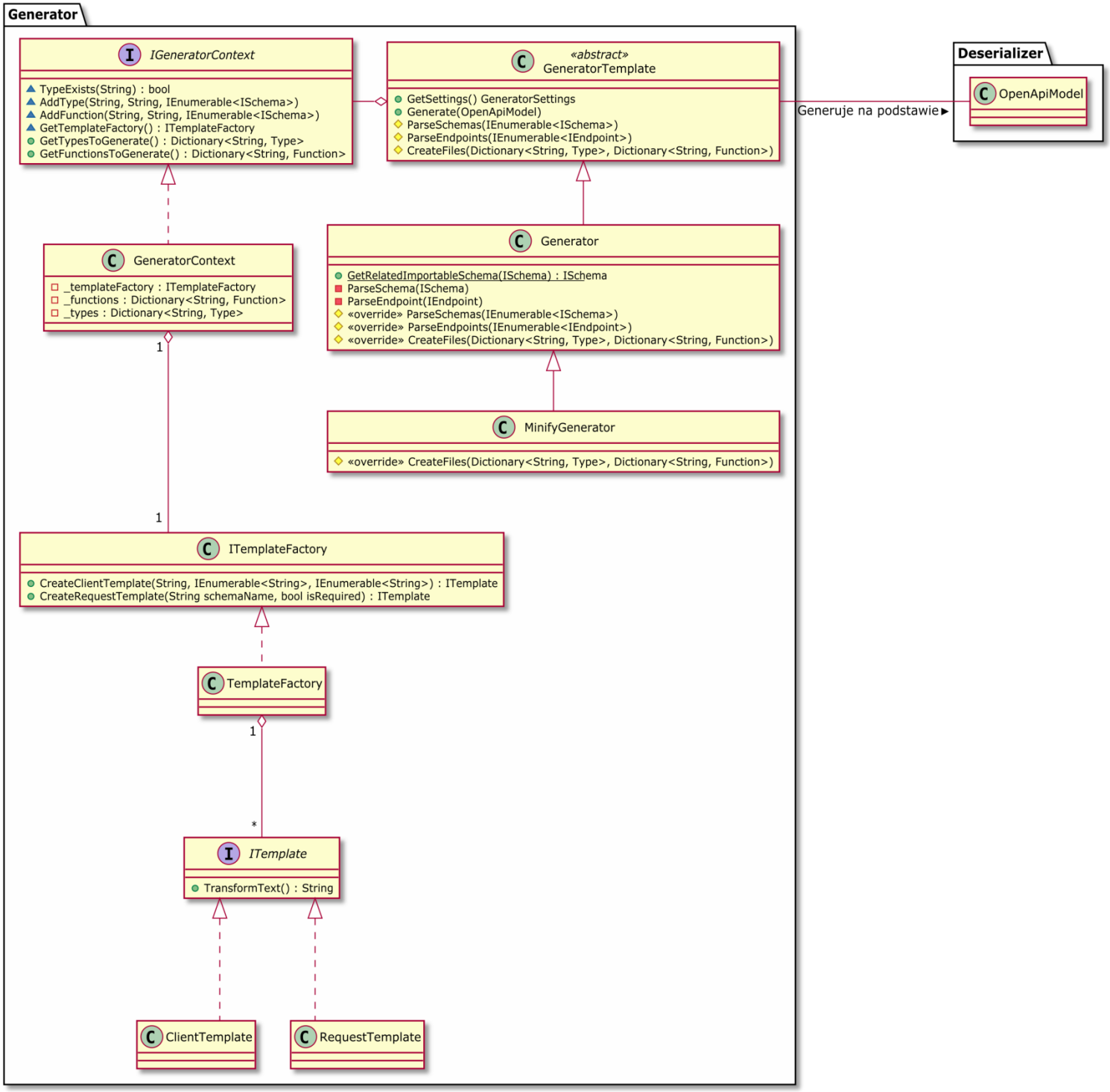
Dodatkowo testy do części serwerowej oraz do generatora kodu zostaną napisane w XUnit. Zostaną również zastosowane niektóre z wzorców projektowych (takie jak wstrzykiwanie zależności, komendy czy fabryka abstrakcyjna). Jako główną bazę danych wybraliśmy bazę MySql. Będziemy ją używać do przechowywania użytkowników. Jednocześnie będzie ona wspierane przez bazę Redisową, która będzie przechowywać cache serwera. Obie bazy zostaną postawione za pomocą Dockerowego containera.

## Diagram przypadków użycia:

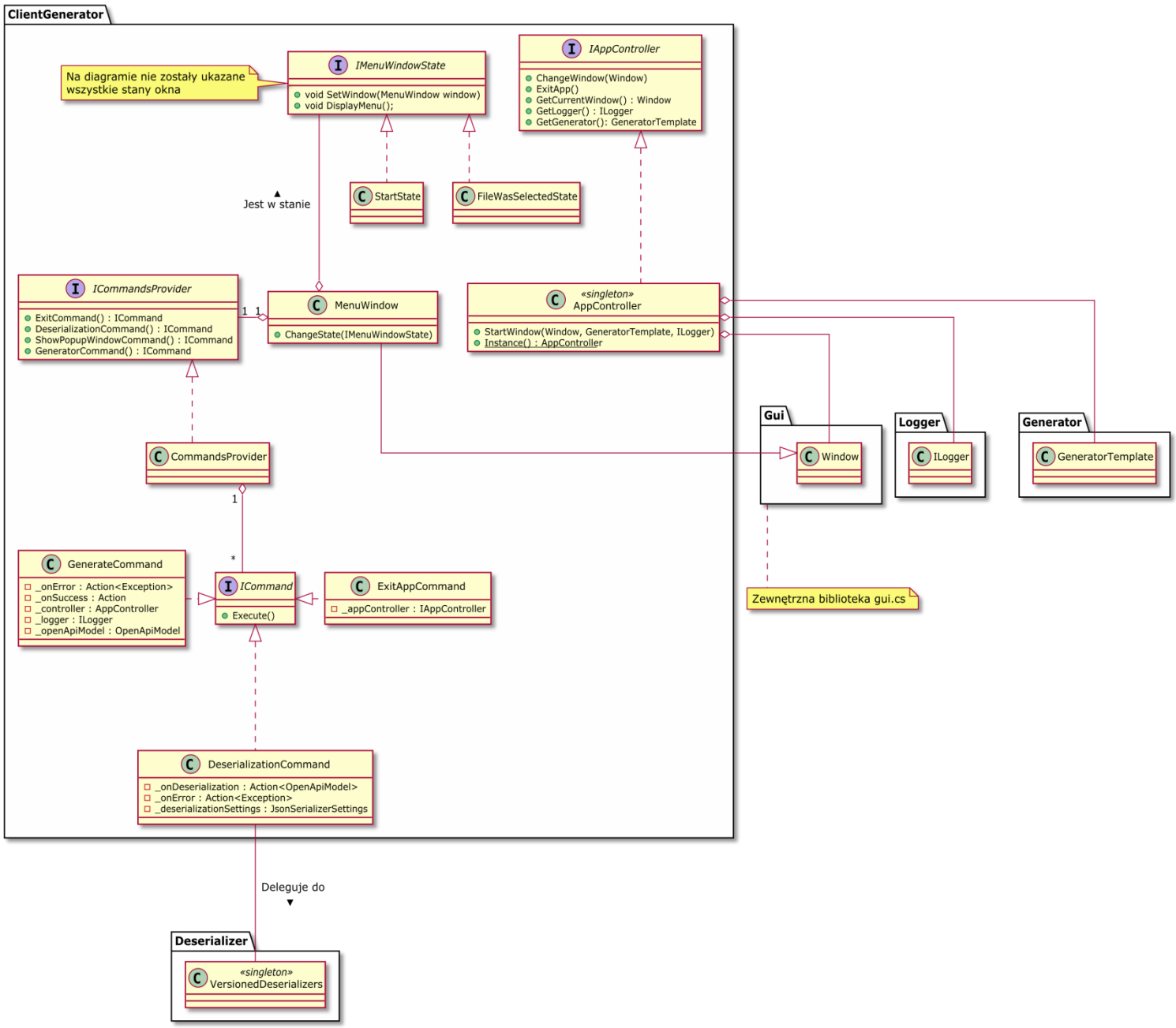


## Diagramy klas:

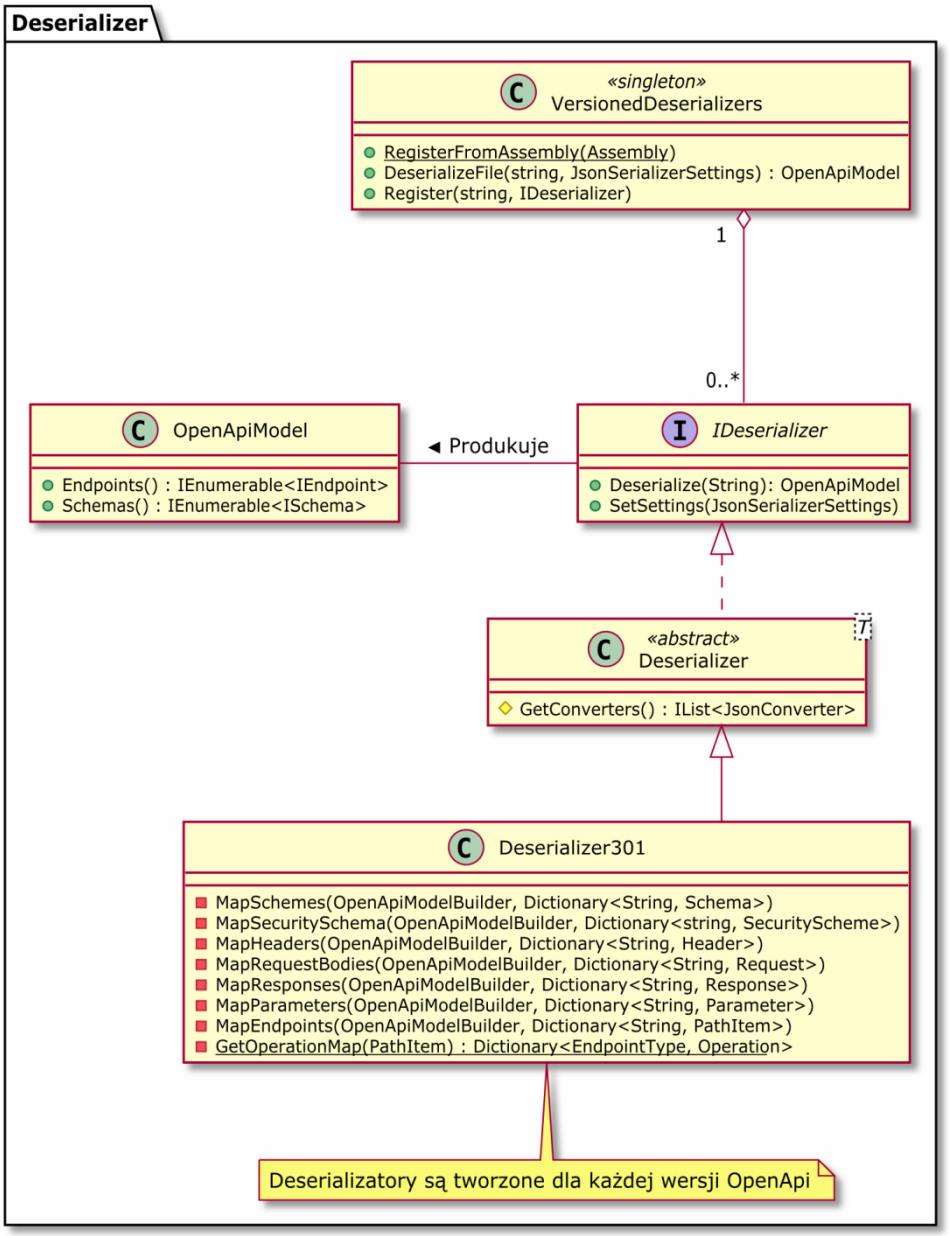
### 1) *moduł Generator:*



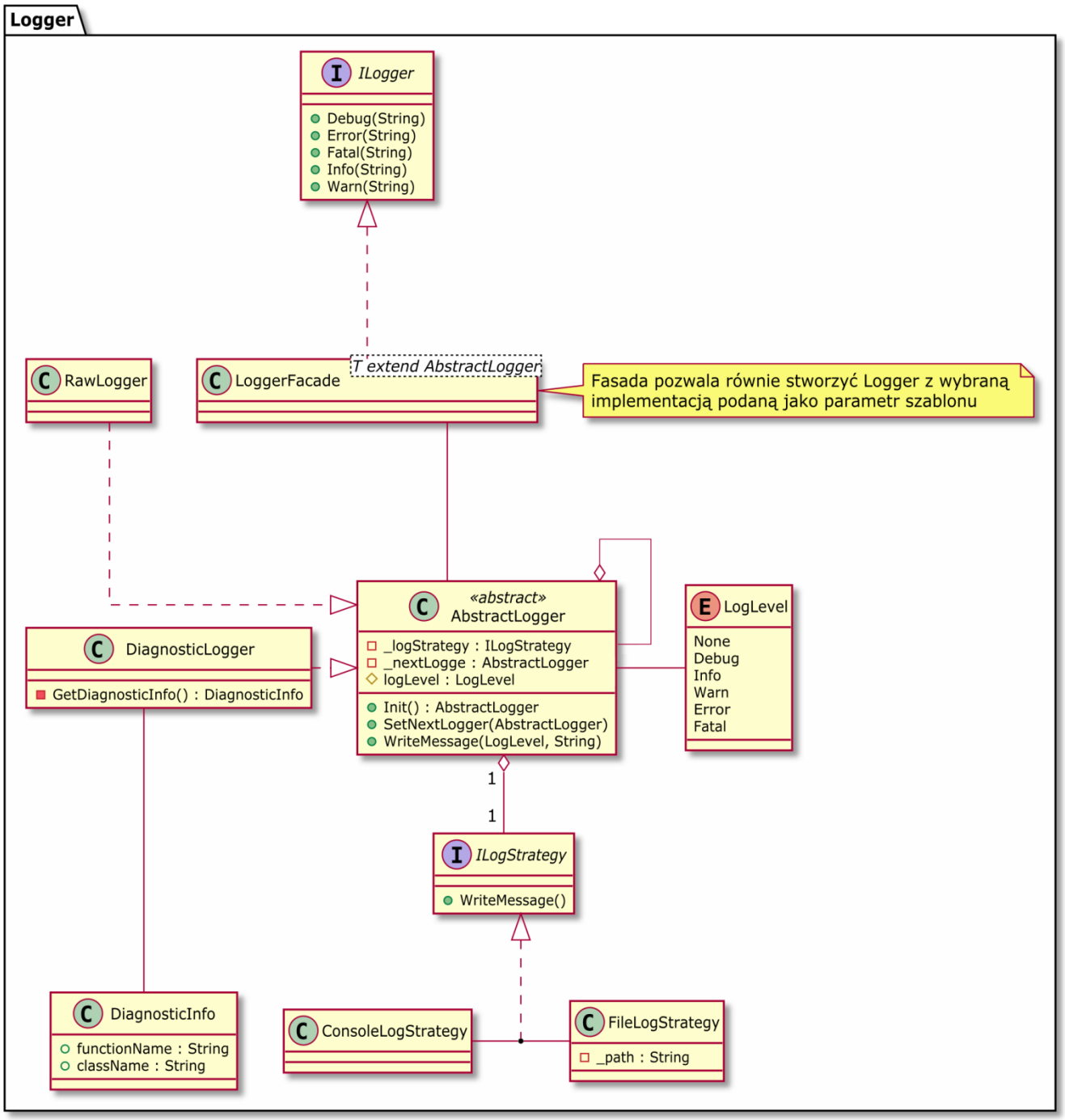
**2)** *moduł ClientGenerator:*



**3)** *moduł Deserializer.*

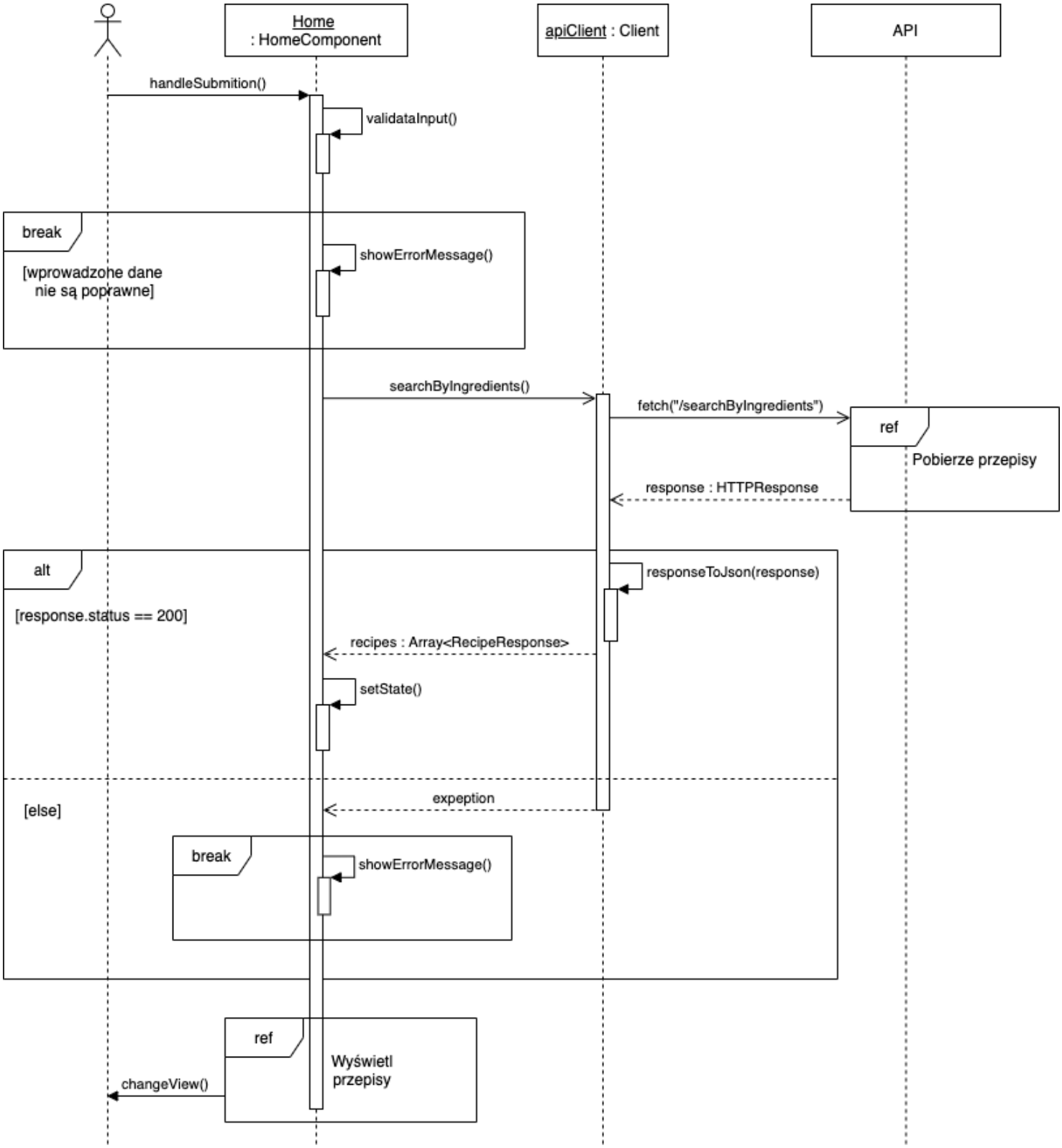


### *4) moduł Logger*

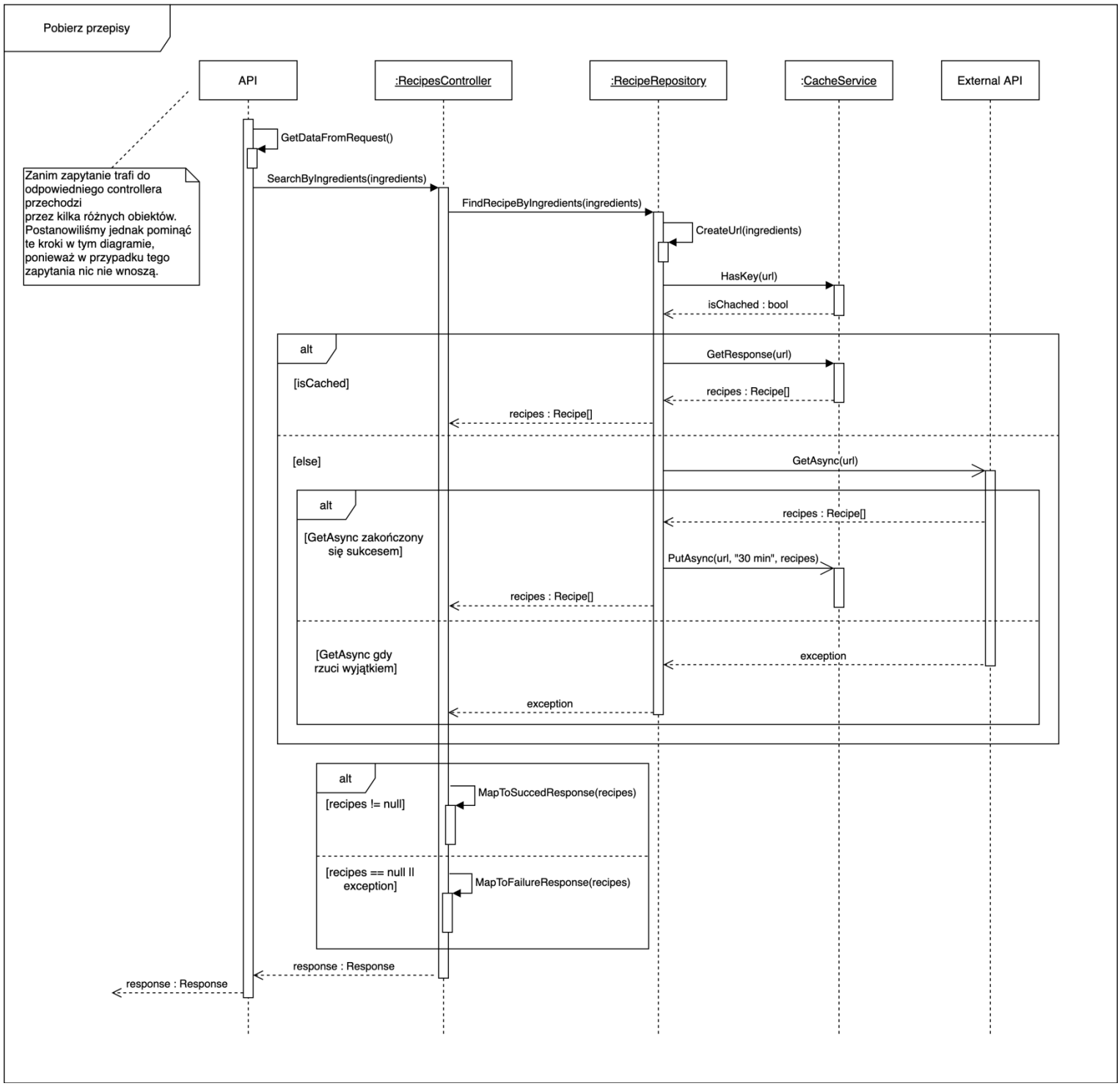


## Diagramy sekwencji:

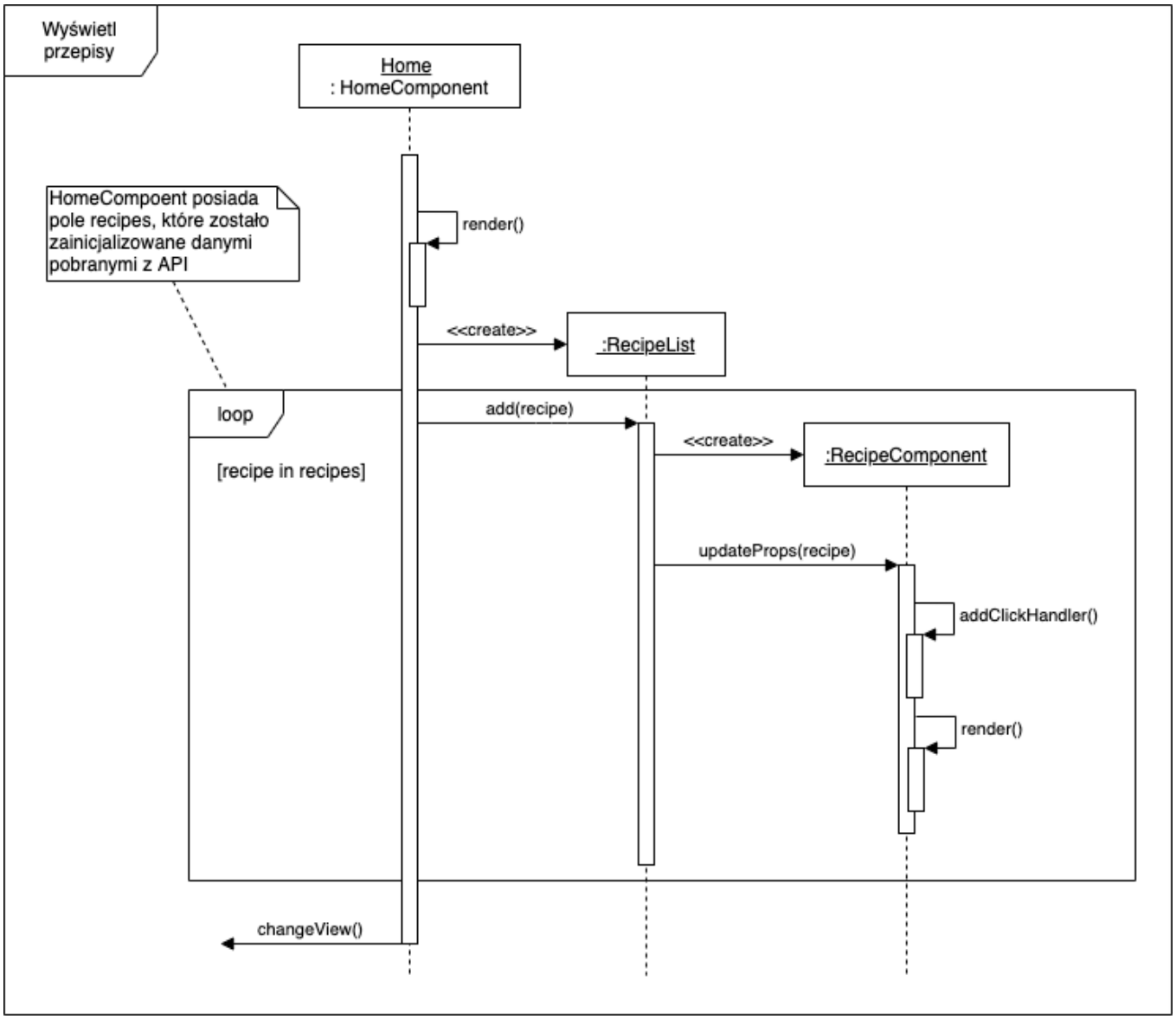
***1)*** *przypadek użycia ‘Wyszukaj przepis’:*



***2)*** *przypadek użycia ‘Pobierz przepisy’:*



***3)*** *przypadek użycia ‘Wyświetl przepisy’:*



## Diagram stanów menu głównego do tworzenia modelu ‘Open Api’ i generowania kodu klienta: