# *Zadanie 2: Specyfikacja wymagań dla aplikacji Giftify*

## Opis ogólny

Giftify to platforma sprzedażowa oferująca certyfikaty prezentowe w różnych kategoriach, takich jak SPA, moda, kulinaria, aktywności i rozwój osobisty. Aplikacja umożliwia użytkownikom wygodne przeglądanie ofert, filtrowanie certyfikatów według preferencji oraz zarządzanie koszykiem zakupów.

## Wymagania funkcjonalne

1. ***Przeglądanie katalogu certyfikatów***
   * Użytkownicy mogą przeglądać dostępne certyfikaty w różnych kategoriach (np. SPA, Moda, Kulinarne, Sport) oraz filtrować je według podstawowych kryteriów, takich jak cena, kategoria, lokalizacja oraz popularność.
2. ***Wyświetlanie szczegółów certyfikatu***
   * Użytkownik ma możliwość obejrzenia szczegółowych informacji o wybranym certyfikacie, w tym opisu, warunków realizacji oraz sugestii podobnych certyfikatów.
3. ***Dodawanie do koszyka***
   * Użytkownik może dodawać certyfikaty do koszyka. Aplikacja będzie przechowywać stan koszyka na czas trwania sesji użytkownika.
4. ***Zarządzanie koszykiem***
   * Użytkownik może dodawać i usuwać certyfikaty z koszyka oraz przeglądać łączny koszt zakupów. Finalizacja zakupów nie obejmuje integracji płatności, jednak aplikacja powinna umożliwiać przechowywanie informacji o wybranych certyfikatach.
5. ***Powiadomienia o nowościach i dostępności***
   * Użytkownicy mogą otrzymywać powiadomienia o nowych dostępnych certyfikatach w kategoriach, które ich interesują.
6. ***Zarządzanie ofertą przez administratora***
   * Administrator ma możliwość dodawania, usuwania oraz edycji certyfikatów dostępnych w katalogu. Zmiany te są od razu widoczne dla użytkowników przeglądających ofertę.

## Wymagania niefunkcjonalne

1. ***Przechowywanie danych w pamięci aplikacji***
   * Dane o certyfikatach i koszyku będą przechowywane w pamięci operacyjnej, bez korzystania z zewnętrznej bazy danych.
2. ***Skalowalność operacyjna***

* Architektura aplikacji powinna umożliwiać łatwe rozszerzenie o nowe funkcje i wzorce projektowe bez zakłócania istniejących funkcjonalności, aby wspierać przyszły rozwój i zwiększenie liczby użytkowników.

1. ***Interfejs przyjazny dla użytkownika***

* Aplikacja powinna posiadać intuicyjny interfejs z przyjaznym UX i dostosowanym UI, aby użytkownicy mogli łatwo nawigować i efektywnie korzystać z funkcji aplikacji. W szczególności, kluczowe funkcje (takie jak filtry i koszyk) powinny być łatwo dostępne.

1. ***Szybkość i responsywność***

* System ma zapewniać płynne działanie na urządzeniach mobilnych i desktopowych, a interfejs użytkownika powinien być prosty i intuicyjny.

## Potencjalne wzorce projektowe

* **Factory Method** – do generowania instancji różnych typów certyfikatów, np. certyfikaty SPA, kulinarne, sportowe. Ułatwia to dodawanie kolejnych kategorii oraz modyfikację poszczególnych atrybutów certyfikatów.
* **Singleton** – do zarządzania koszykiem, zapewniając dostęp do jednej, wspólnej instancji koszyka na sesję użytkownika.
* **Strategy** – do zastosowania różnych metod filtrowania i rekomendacji certyfikatów, co pozwala na elastyczność logiki filtrowania według preferencji użytkownika.
* **Observer** – do powiadamiania użytkowników o nowościach w wybranych kategoriach oraz o aktualizacjach oferty.
* **Decorator** – do dodawania opcjonalnych cech certyfikatów (np. promocje lub tagi „premium”) bez zmiany podstawowej struktury certyfikatu.
* **Facade** – do uproszczenia komunikacji między komponentami aplikacji, takimi jak koszyk i katalog certyfikatów, co ułatwia rozwój i obsługę różnych funkcji aplikacji.