Załącznik A

Szczegółowe scenariusze użycia

Poniżej przedstawiono szczegółowe opisy scenariuszy użycia systemu odpowiadających przypadkom użycia zaprezentowanym na diagramie w rozdziale 3.3. Scenariusze zostały opisane zgodnie z notacją: Aktor, Warunki początkowe, Przebieg scenariusza głównego, Warunki końcowe.

Scenariusz 1: Przeglądanie i wyszukiwanie produktów

Aktor: Gość (użytkownik niezalogowany)
Warunki początkowe: Użytkownik znajduje się na stronie głównej aplikacji
Przebieg scenariusza głównego:

1. Użytkownik otwiera stronę główną aplikacji w przeglądarce
2. Użytkownik otwiera pasek wyszukiwania z standardową opcją
3. Użytkownik wpisuje szukaną frazę
4. System sortuje produkty według wartości `sentiment_score` malejąco
5. Użytkownik przegląda listę produktów
6. Użytkownik klika na wybrany produkt
7. System wyświetla szczegółową stronę produktu z opisem, specyfikacjami, zdjęciami i opiniami

Warunki końcowe: Użytkownik widzi szczegółowe informacje o wybranym produkcie

Scenariusz 2: Dodawanie produktów do koszyka

Aktor: Gość (użytkownik niezalogowany)
Warunki początkowe: Użytkownik przegląda szczegółową stronę produktu (laptop)
Przebieg scenariusza głównego:

1. Użytkownik klika przycisk „Dodaj do koszyka”
2. System dodaje produkt do koszyka i wyświetla powiadomienie o sukcesie
3. Użytkownik klika ikonę koszyka w nawigacji
4. System wyświetla widok koszyka z dodanymi produktami
5. System automatycznie wywołuje algorytm Apriori dla produktów w koszyku
6. System wyświetla sekcję „Frequently Bought Together” z trzema produktami rekomendowanymi przez algorytm Apriori (torba na laptopa, mysz bezprzewodowa, hub USB-C)

7. Każda rekomendacja zawiera metryki: lift, confidence oraz przycisk „Dodaj do koszyka”
8. Użytkownik przegląda rekomendacje
9. Użytkownik klika „Dodaj do koszyka” przy torbie i myszy
10. System dodaje wybrane produkty do koszyka i aktualizuje sumę całkowitą

Warunki końcowe: Koszyk zawiera laptop, torbę i mysz, wyświetlona jest zaktualizowana suma

Scenariusz 3: Rejestracja nowego użytkownika

Aktor: Gość (użytkownik niezalogowany)
Warunki początkowe: Użytkownik znajduje się na stronie głównej
Przebieg scenariusza głównego:

1. Użytkownik klika przycisk „Zarejestruj się” w nawigacji
2. System wyświetla formularz rejestracji
3. Użytkownik wypełnia pola: imię, nazwisko, adres e-mail, hasło, potwierdzenie hasła
4. Użytkownik klika przycisk „Zarejestruj”
5. System waliduje dane: sprawdza czy e-mail jest unikalny, czy hasła się zgadzają, czy hasło spełnia wymagania bezpieczeństwa
6. System tworzy nowe konto użytkownika w bazie danych
7. System wyświetla komunikat o pomyślnej rejestracji
8. System automatycznie loguje użytkownika i generuje token JWT
9. System przekierowuje użytkownika do panelu klienta

Warunki końcowe: Użytkownik jest zalogowany i posiada aktywne konto w systemie

Scenariusz 4: Logowanie do systemu

Aktor: Gość (użytkownik niezalogowany, posiadający konto)
Warunki początkowe: Użytkownik znajduje się na stronie głównej
Przebieg scenariusza głównego:

1. Użytkownik klika przycisk „Zaloguj się” w nawigacji
2. System wyświetla formularz logowania
3. Użytkownik wpisuje adres e-mail i hasło
4. Użytkownik klika przycisk „Zaloguj”

5. System weryfikuje dane logowania w bazie danych
6. System generuje token JWT i zwraca go do klienta
7. Frontend zapisuje token w localStorage przeglądarki
8. System przekierowuje użytkownika do panelu klienta

Warunki końcowe: Użytkownik jest zalogowany, posiada ważny token JWT

Scenariusz 5: Śledzenie statusu zamówienia

Aktor: Klient (użytkownik zalogowany)
Warunki początkowe: Użytkownik jest zalogowany, posiada co najmniej jedno zamówienie
Przebieg scenariusza głównego:

1. Użytkownik otwiera panel klienta
2. Użytkownik klika zakładkę „My Orders"
3. System pobiera wszystkie zamówienia użytkownika z bazy danych
4. System wyświetla listę zamówień posortowaną od najnowszych
5. Dla każdego zamówienia wyświetlane są: numer zamówienia, status (oczekujące/w realizacji/zakończone/anulowane)

Warunki końcowe: Użytkownik widzi szczegółowe informacje o statusie wybranego zamówienia

Scenariusz 6: Wystawianie opinii o produkcie

Aktor: Klient (użytkownik zalogowany)
Warunki początkowe: Użytkownik posiada zakończone zamówienie zawierające produkt
Przebieg scenariusza głównego:

1. Użytkownik otwiera panel klienta, zakładkę „My Orders"
2. Użytkownik znajduje zakończone zamówienie
3. Użytkownik klika przycisk „Add Review" przy zakupionym produkcie
4. System wyświetla formularz opinii
5. Użytkownik wybiera ocenę gwiazdkową (1-5 gwiazdek)
6. Użytkownik wpisuje treść tekstową opinii
7. Użytkownik klika „Wyślij opinię"
8. System zapisuje opinię w bazie danych z powiązaniem: użytkownik, produkt, zamówienie

9. System automatycznie wywołuje algorytm analizy sentymentu dla nowej opinii
10. Algorytm tokenizuje tekst, wyszukuje słowa w słowniku pozytywnym/negatywnym
11. Algorytm oblicza sentiment_score dla opinii zgodnie ze wzorem (2)
12. System przelicza zagregowany sentyment produktu zgodnie ze wzorem (3) (wieloźródłowa agregacja z 5 źródeł)
13. System aktualizuje pole `sentiment_score` w rekordzie produktu
14. System wyświetla komunikat o pomyślnym dodaniu opinii
15. System odświeża widok produktu - zaktualizowany sentyment jest widoczny

Warunki końcowe: Opinia została dodana, sentyment produktu zaktualizowany

Scenariusz 7: Zarządzanie kontem użytkownika

Aktor: Klient (użytkownik zalogowany)
Warunki początkowe: Użytkownik jest zalogowany
Przebieg scenariusza głównego:

1. Użytkownik otwiera panel klienta
2. Użytkownik klika zakładkę „Konto"
3. System wyświetla formularz z danymi użytkownika: imię, nazwisko, e-mail
4. Użytkownik edytuje wybrane pola (np. zmienia nazwisko)
5. Użytkownik klika „Zapisz zmiany"
6. System waliduje dane (sprawdza czy e-mail jest unikalny, jeśli został zmieniony)
7. System aktualizuje rekord użytkownika w bazie danych
8. System wyświetla komunikat o pomyślnym zapisaniu zmian

Warunki końcowe: Dane użytkownika zostały zaktualizowane

Scenariusz 8: Administrator zarządza produktami

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, posiada uprawnienia administratora
Przebieg scenariusza głównego - dodawanie produktu:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Products"
3. System wyświetla listę wszystkich produktów

4. Administrator klika przycisk „Add New Product”
5. System wyświetla formularz dodawania produktu
6. Administrator wypełnia pola: nazwa, opis, cena, kategorie, specyfikacje techniczne, przesyła zdjęcie
7. Administrator klika „Zapisz produkt”
8. System waliduje dane (sprawdza czy wszystkie wymagane pola są wypełnione)
9. System tworzy nowy rekord produktu w bazie danych
10. System automatycznie wywołuje algorytm analizy sentymentu dla nowego produktu
11. Algorytm oblicza sentiment_score na podstawie nazwy i opisu (brak jeszcze opinii)
12. System zapisuje obliczony sentyment w rekordzie produktu
13. System wyświetla komunikat o pomyślnym dodaniu produktu
14. System odświeża listę produktów

Warunki końcowe: Nowy produkt został dodany do katalogu z obliczonym sentymentem

Scenariusz 9: Administrator zarządza zamówieniami

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Orders”
3. System wyświetla listę wszystkich zamówień w systemie
4. Dla każdego zamówienia wyświetlane są: numer, aktualny status
5. Administrator klika na wybrane zamówienie ze statusem „oczekujące”
6. Administrator klika przycisk „Zmień status”
7. System wyświetla listę dostępnych statusów (w realizacji, zakończone, anulowane)
8. Administrator wybiera „w realizacji”
9. System aktualizuje status zamówienia w bazie danych
10. System zapisuje zdarzenie zmiany statusu z timestampem
11. System wyświetla komunikat o pomyślnej zmianie statusu

12. Użytkownik właściciel zamówienia widzi zaktualizowany status w swoim panelu

Warunki końcowe: Status zamówienia został zmieniony, użytkownik ma dostęp do aktualnej informacji

Scenariusz 10: Administrator zarządza użytkownikami

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Users"
3. System wyświetla listę wszystkich użytkowników
4. Dla każdego użytkownika wyświetlane są: ID, imię, nazwisko, e-mail, rola, data rejestracji
5. Administrator może filtrować użytkowników według roli (klient/administrator)
6. Administrator klika na wybranego użytkownika
7. System wyświetla szczegóły użytkownika: profil, historia zamówień, statystyki
8. Administrator może zmienić rolę użytkownika (nadać uprawnienia administratora) lub usunąć konto

Warunki końcowe: Administrator ma pełny wgląd w dane użytkowników i może zarządzać uprawnieniami

Scenariusz 11: Administrator przegląda statystyki i dashboard

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. System automatycznie wyświetla dashboard ze statystykami
3. Dashboard zawiera następujące kluczowe metryki:
    - Całkowita liczba produktów w katalogu
    - Całkowita liczba zarejestrowanych użytkowników
    - Całkowita liczba zamówień (wszystkie statusy)
    - Całkowity przychód (suma wartości zamówień zakończonych)
4. System wyświetla wykresy:
    - Wykres słupkowy: miesięczny obrót (ostatnie 12 miesięcy)

- Wykres kołowy: rozkład zamówień według kategorii produktów
- Wykres liniowy: liczba nowych użytkowników (ostatnie 30 dni)

5. Administrator klika sekcję „Statistics" dla bardziej szczegółowych analiz
6. System wyświetla widok do zmiany rekomentacji, odświeżanie reguł asocjacyjnych

Warunki końcowe: Administrator ma pełny wgląd w kluczowe wskaźniki biznesowe systemu

Scenariusz 12: Administrator debuguje algorytmy rekomendacji

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, algorytmy rekomendacji zostały przeliczone
Przebieg scenariusza głównego - Collaborative Filtering:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Debug"
3. Administrator wybiera zakładkę „Collaborative Filtering"
4. System wyświetla panel debugowania Collaborative Filtering zawierający:
   - Wzór matematyczny Adjusted Cosine Similarity
   - Statystyki: liczba par produktów, średnie podobieństwo, czas ostatniego przeliczenia
   - Tabelę z przykładowymi podobieństwami (product_a, product_b, similarity_score)
5. Administrator może wpisać ID produktu i zobaczyć top 10 najbardziej podobnych produktów
6. System pobiera dane z tabeli ProductSimilarity i wyświetla wyniki

Warunki końcowe: Administrator zweryfikował poprawność działania algorytmu Collaborative Filtering

Scenariusz 13: Przeglądanie rekomendacji na stronie głównej

Aktor: Klient (użytkownik zalogowany)
Warunki początkowe: Użytkownik jest zalogowany, posiada historię zakupów
Przebieg scenariusza głównego:

1. Użytkownik loguje się do systemu
2. System przekierowuje użytkownika do strony głównej

3. System wywołuje algorytm Collaborative Filtering dla zalogowanego użytkownika
4. Algorytm analizuje historię zakupów użytkownika
5. Algorytm pobiera produkty podobne do wcześniej zakupionych (na podstawie macierzy podobieństw)
6. System wyświetla sekcję „Rekomendowane dla Ciebie" z 6 produktami
7. Użytkownik przegląda rekomendacje
8. Użytkownik klika na wybrany produkt
9. System wyświetla szczegółową stronę produktu

Warunki końcowe: Użytkownik widzi spersonalizowane rekomendacje oparte na swojej historii zakupów

Scenariusz 14: Przeglądanie podobnych produktów (Content-Based Filtering)

Aktor: Klient lub Gość
Warunki początkowe: Użytkownik przegląda szczegółową stronę produktu (np. laptop Dell XPS 15)
Przebieg scenariusza głównego:

1. Użytkownik otwiera stronę szczegółową produktu
2. System automatycznie wywołuje endpoint `GET /api/recommendations/content-based/?p`
3. Backend wywołuje metodę `ContentBasedAPI` z parametrem `product_id`
4. System wykonuje zapytanie do tabeli `ProductSimilarity` filtrując po `product1_id = {id}` oraz `type='content_based'`
5. System sortuje wyniki malejąco według `similarity_score`
6. System pobiera Top 10 najbardziej podobnych produktów z bazy danych
7. Backend zwraca JSON z listą produktów zawierającą: `product_id`, `name`, `price`, `similarity_score`
8. Frontend wyświetla sekcję „Podobne produkty"z kafelkami produktów
9. Każdy kafelek zawiera zdjęcie, nazwę, cenę oraz wskaźnik podobieństwa (np. „85% podobny")
10. Użytkownik przegląda rekomendacje
11. Użytkownik klika na wybrany podobny produkt
12. System przekierowuje do strony szczegółowej wybranego produktu

Warunki końcowe: Użytkownik widzi produkty podobne do oglądanego, obliczone na podstawie cech (kategorie 40%, tagi 30%, cena 20%, słowa kluczowe 10%)

Scenariusz 15: Generowanie macierzy podobieństw Content-Based Filtering

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, katalog zawiera produkty
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Admin Tools"
3. Administrator wybiera zakładkę „Content-Based Filtering"
4. System wyświetla panel zarządzania CBF z przyciskiem „Generate Similarity Matrix"
5. Administrator klika „Generate Similarity Matrix"
6. System wywołuje endpoint `POST /api/admin/content-based/generate`
7. Backend uruchamia algorytm `ContentBasedRecommendationEngine.generate_similariti`
8. System pobiera wszystkie produkty z bazy danych
9. Dla każdego produktu system ekstraktuje cechy:
    - Kategorie (waga 40%) - konwersja do wektora one-hot
    - Tagi (waga 30%) - ekstrakcja unikalnych tagów
    - Przedział cenowy (waga 20%) - normalizacja do zakresu [0,1]
    - Słowa kluczowe z opisu (waga 10%) - tokenizacja i TF-IDF
10. System oblicza podobieństwo cosinusowe między każdą parą produktów zgodnie ze wzorem Weighted Cosine Similarity
11. System filtruje pary z podobieństwem $> 20\%$ (próg minimalny)
12. System usuwa stare rekordy z tabeli `ProductSimilarity` dla `type='content_based'`
13. System zapisuje nowe podobieństwa do tabeli `ProductSimilarity`
14. System wyświetla komunikat o sukcesie: „Wygenerowano 1247 podobieństw dla 500 produktów w czasie 58 sekund"
15. Administrator widzi zaktualizowane statystyki: liczba par produktów, średnie podobieństwo, timestamp ostatniego przeliczenia

Warunki końcowe: Macierz podobieństw została wygenerowana i zapisana w bazie danych, rekomendacje CBF są aktualne

Scenariusz 16: Debugowanie algorytmu Content-Based Filtering

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, macierz podobieństw została wygenerowana
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Debug"
3. Administrator wybiera zakładkę „Content-Based Debug"
4. System wyświetla panel debugowania CBF zawierający:
   - Wzór matematyczny Weighted Cosine Similarity z opisem wag cech
   - Statystyki globalne: liczba produktów w katalogu, liczba wygenerowanych par, średnie podobieństwo, czas ostatniego przeliczenia
   - Histogram rozkładu podobieństw (przedziały: 20-40%, 40-60%, 60-80%, 80-100%)
5. Administrator wprowadza ID produktu w pole „Product ID for analysis"
6. Administrator klika „Analyze Product"
7. System wywołuje endpoint `GET /api/admin/content-based/debug/?product_id={id}`
8. System pobiera wybrany produkt oraz jego Top 10 najbardziej podobnych produktów
9. System wyświetla szczegółową analizę dla wybranego produktu:
   - Podstawowe dane: nazwa, kategorie, tagi, cena
   - Ekstrahowane cechy: wektor kategorii, lista tagów, znormalizowana cena, słowa kluczowe
   - Tabelę Top 10 podobnych produktów z kolumnami: ID, nazwa, similarity_score, wspólne kategorie, wspólne tagi
10. Administrator analizuje wyniki i weryfikuje poprawność podobieństw
11. Administrator może wyeksportować dane do CSV klikając „Export to CSV"

Warunki końcowe: Administrator zweryfikował poprawność działania algorytmu CBF dla wybranego produktu

Scenariusz 17: Przeglądanie rekomendacji Fuzzy Logic w panelu klienta

Aktor: Klient (użytkownik zalogowany)
Warunki początkowe: Użytkownik jest zalogowany, posiada historię zakupów
Przebieg scenariusza głównego:

1. Użytkownik loguje się do systemu
2. Użytkownik otwiera panel klienta
3. Użytkownik klika zakładkę „Fuzzy Logic Recommendations"
4. System wywołuje endpoint `GET /api/fuzzy-logic-recommendations/?limit=10`
5. Backend wywołuje `FuzzyLogicAPI` z parametrem `user_id` (z tokenu JWT)
6. System inicjalizuje silnik rozmyty `FuzzyEngine.initialize_fuzzy_system(user_id)`
7. System buduje profil użytkownika `FuzzyUserProfile`:
   - Oblicza parametr `price_sensitivity` na podstawie średniej ceny zakupionych produktów
   - Ekstraktuje preferowane kategorie z historii zakupów (top 3 kategorie)
8. System pobiera wszystkie produkty z bazy danych z metrykami: cena, średni rating, liczba zamówień
9. Dla każdego produktu system wykonuje wnioskowanie rozmyte:
   - Fuzzyfikacja - oblicza przynależność ceny, jakości i popularności do zbiorów rozmytych (cheap/medium/expensive, low/medium/high)
   - Aplikacja 6 reguł IF-THEN typu Mamdani (R1: preferred category match, R2: quality-price balance, R3: price sensitive match, R4: premium match, R5: popular high quality, R6: budget friendly)
   - Agregacja wyników reguł z wagami (R1: 0.9, R2: 0.85, R3: 0.6, R4: 0.75, R5: 0.8, R6: 0.7)
   - Defuzzyfikacja - oblicza końcowy `fuzzy_score` metodą średniej ważonej
10. System sortuje produkty malejąco według `fuzzy_score`
11. System zwraca JSON z Top 10 rekomendacji zawierający: `product_id`, `name`, `price`, `fuzzy_score`, `activated_rules`
12. Frontend wyświetla trzy zakładki:
    - „Fuzzy Recommendations lista Top 10 produktów z fuzzy_score i aktywowanymi regułami
    - „User Profile wyświetla `price_sensitivity` (np. 60% - moderate) oraz preferowane kategorie
    - „Fuzzy Rules opis 6 reguł IF-THEN z wyjaśnieniem logiki

13. Użytkownik przegląda rekomendacje dopasowane do jego profilu cenowego
14. Użytkownik klika przycisk „View Rule Activations"przy wybranym produkcie
15. System wyświetla szczegóły aktywacji reguł: które reguły zostały aktywowane, z jaką siłą (α_i), jaki był ich wkład w końcowy wynik

Warunki końcowe: Użytkownik widzi spersonalizowane rekomendacje dopasowane do jego wrażliwości cenowej oraz preferowanych kategorii z pełną przejrzystością procesu decyzyjnego

Scenariusz 18: Wyszukiwanie tolerujące błędy (Fuzzy Search)

Aktor: Klient lub Gość
Warunki początkowe: Użytkownik znajduje się na stronie głównej
Przebieg scenariusza głównego:

1. Użytkownik otwiera pasek wyszukiwania
2. Użytkownik przełącza opcję wyszukiwania na „Fuzzy Search"
3. System wyświetla suwak „Fuzzy Threshold"(domyślnie 0.5 = 50%)
4. Użytkownik wpisuje frazę z literówką: „loptap"(zamiast „laptop")
5. Użytkownik ustawia próg podobieństwa na 0.3 (30%)
6. Użytkownik klika przycisk „Search"
7. System wywołuje endpoint `GET /api/search/fuzzy/?query=loptap&threshold=0.3`
8. Backend iteruje po wszystkich produktach w katalogu
9. Dla każdego produktu system oblicza odległość Levenshteina między zapytaniem a nazwą produktu:
    - Algorytm rekurencyjny oblicza minimalną liczbę operacji edycji (wstawienie, usunięcie, zamiana znaku)
    - System normalizuje odległość do zakresu [0,1] dzieląc przez maksymalną długość
    - Oblicza `fuzzy_match_score` = 1 - normalized_distance
10. System filtruje produkty z `fuzzy_match_score` ≥ 0.3 (próg użytkownika)
11. System sortuje wyniki malejąco według `fuzzy_match_score`
12. Backend zwraca JSON z produktami: `product_id`, `name`, `price`, `fuzzy_match_score`
13. Frontend wyświetla wyniki wyszukiwania:
    - Laptop Dell XPS 15 - 86% match (zawiera „Laptop")
    - Laptop HP Pavilion - 86% match (zawiera „Laptop")
    - Desktop Hub USB-C - 45% match (częściowe dopasowanie)

14. Użytkownik widzi produkty mimo błędu w zapytaniu
15. Użytkownik klika na wybrany produkt

Warunki końcowe: Użytkownik znalazł poszukiwane produkty mimo literówki dzięki algorytmowi Levenshteina

Scenariusz 19: Debugowanie algorytmu Fuzzy Logic

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, system posiada użytkowników z historią zakupów
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Debug"
3. Administrator wybiera zakładkę „Fuzzy Logic Debug"
4. System wyświetla widok ogólny panelu debugowania zawierający:
   - Szczegóły algorytmu: metoda Mamdani Fuzzy Inference, liczba reguł (6), T-norma (min), T-conorma (max)
   - Definicje funkcji przynależności dla price (cheap: 0-500 PLN, medium: 300-1500 PLN, expensive: 1000+ PLN)
   - Definicje funkcji przynależności dla quality (low: 0-2.5, medium: 2.0-4.0, high: 3.5-5.0)
   - Definicje funkcji przynależności dla popularity (low: 0-10 zamówień, medium: 5-50, high: 30+ zamówień)
   - Listę profili użytkowników z parametrem price_sensitivity
5. Administrator wprowadza ID produktu w pole „Product ID for detailed analysis"
6. Administrator klika „Analyze Product"
7. System wywołuje endpoint `GET /api/admin/fuzzy-debug/?product_id={id}`
8. System pobiera wybrany produkt oraz wykonuje szczegółową ewaluację rozmytą
9. System wyświetla widok szczegółowy dla produktu:
   - Dane produktu: nazwa, cena, średni rating, liczba zamówień
   - Wartości fuzzyfikacji:
     - Price: cheap=0.3, medium=0.7, expensive=0.0
     - Quality: low=0.0, medium=0.2, high=0.8
     - Popularity: low=0.0, medium=0.6, high=0.4
   - Aktywacja wszystkich 6 reguł z siłą aktywacji α_i:

- R1 (Preferred Category Match): $\alpha_1 = 0.85$, waga = 0.9, wkład = 0.765
- R2 (Quality-Price Balance): $\alpha_2 = 0.7$, waga = 0.85, wkład = 0.595
- R3 (Price Sensitive Match): $\alpha_3 = 0.3$, waga = 0.6, wkład = 0.18
- R4 (Premium Match): $\alpha_4 = 0.0$, waga = 0.75, wkład = 0.0
- R5 (Popular High Quality): $\alpha_5 = 0.6$, waga = 0.8, wkład = 0.48
- R6 (Budget Friendly): $\alpha_6 = 0.2$, waga = 0.7, wkład = 0.14

- Agregacja: suma wkładów = 2.16, fuzzy_score = 2.16 / 4.6 = 0.47 (47%)

10. Administrator analizuje które reguły dominują dla danego produktu
11. Administrator może dostroić wagi reguł klikając „Edit Rule Weights"
12. System zapisuje nowe wagi w konfiguracji

Warunki końcowe: Administrator zweryfikował proces wnioskowania rozmytego i zoptymalizował wagi reguł

Scenariusz 20: Przeglądanie rekomendacji probabilistycznych w panelu klienta

Aktor: Klient (użytkownik zalogowany)
Warunki początkowe: Użytkownik jest zalogowany, posiada historię zakupów, modele probabilistyczne zostały wytrenowane
Przebieg scenariusza głównego:

1. Użytkownik loguje się do systemu
2. Użytkownik otwiera panel klienta
3. Użytkownik klika zakładkę „Probabilistic Recommendations"
4. System wywołuje endpoint `GET /api/probabilistic-recommendations/?user_id={id}`
5. Backend wywołuje `ProbabilisticAPI` z parametrem `user_id`
6. System pobiera historię zamówień użytkownika z bazy danych wywołując `get_user_orders(`
7. System ekstraktuje sekwencję kategorii produktów z zamówień uporządkowanych chronologicznie
8. System wywołuje `MarkovChain.predict_next_products(order_sequence)`:
   - Identyfikuje ostatnią zakupioną kategorię (np. „Laptopy")
   - Odczytuje macierz przejść dla kategorii „Laptopy"
   - Pobiera Top 3 kategorie z najwyższym prawdopodobieństwem przejścia (np. Akcesoria komputerowe: 0.45, Peryferia: 0.30, Torby: 0.15)
   - Dla każdej kategorii pobiera Top produkty
9. System wywołuje `NaiveBayes.predict_purchase_probability(user_profile)`:

- Ekstraktuje 5 cech użytkownika: liczba zamówień, średnia wartość, dni od ostatniego zakupu, ulubiona kategoria, częstotliwość
- Stosuje twierdzenie Bayesa dla klasyfikacji binarnej (kupi/nie kupi)
- Oblicza prawdopodobieństwo zakupu w najbliższym okresie

10. System agreguje wyniki: 60% waga dla Markov, 40% waga dla Naive Bayes
11. Backend zwraca JSON z Top 10 rekomendacji zawierający: `product_id`, `name`, `price`, `markov_probability`, `bayes_probability`, `combined_score`
12. Frontend wyświetla trzy zakładki:
    - „Probabilistic Recommendations lista Top 10 produktów z prawdopodobieństwami
    - „Markov Chain Visualization graf przejść między kategoriami z prawdopodobieństwami
    - „Purchase Probability Analysis wykres prawdopodobieństwa zakupu oraz analiza cech użytkownika
13. Użytkownik przegląda rekomendacje oparte na sekwencjach zakupowych
14. Użytkownik klika zakładkę „Markov Chain Visualization"
15. System wyświetla interaktywny graf pokazujący możliwe przejścia z obecnej kategorii do kolejnych
16. Użytkownik klika zakładkę „Purchase Probability Analysis"
17. System wyświetla:
    - Prawdopodobieństwo zakupu: 73% (high probability)
    - Breakdown cech: liczba zamówień: 12 (pozytywny wpływ +0.15), średnia wartość: 450 PLN (pozytywny +0.10), dni od ostatniego: 15 (pozytywny +0.20)
    - Rekomendacja: „Użytkownik jest aktywny, wysokie prawdopodobieństwo konwersji"

Warunki końcowe: Użytkownik widzi rekomendacje przewidujące jego przyszłe zakupy oparte na łańcuchach Markowa i klasyfikatorze Bayesa

Scenariusz 21: Trenowanie modeli probabilistycznych przez administratora

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, system posiada historię zamówień
Przebieg scenariusza głównego:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Admin Tools"

3. Administrator wybiera zakładkę „Probabilistic Models"
4. System wyświetla panel zarządzania modelami probabilistycznymi
5. Administrator widzi status modeli:
   - Markov Chain: ostatni trening 2024-01-15, liczba stanów: 48 kategorii, wygładzanie Laplace'a: $\alpha = 1.0$
   - Naive Bayes: ostatni trening 2024-01-15, liczba próbek treningowych: 200 zamówień, accuracy: 78%
6. Administrator klika przycisk „Train Markov Chain Model"
7. System wywołuje endpoint `POST /api/admin/probabilistic/train-markov`
8. Backend uruchamia `MarkovChainEngine.train()`:
   - Pobiera wszystkie zamówienia z bazy danych uporządkowane chronologicznie per użytkownik
   - Ekstraktuje sekwencje kategorii dla każdego użytkownika
   - Buduje macierz przejść 48x48 (48 kategorii produktowych)
   - Oblicza prawdopodobieństwa przejść: $P(s_j|s_i) = \frac{count(s_i \to s_j) + \alpha}{\sum_k count(s_i \to s_k) + \alpha \cdot N}$
   - Stosuje wygładzanie Laplace'a ($\alpha = 1.0$) zapobiegające zerowemu prawdopodobieństwu
   - Zapisuje macierz w cache Django z timeout 24h
9. System wyświetla komunikat: „Markov Chain trained successfully. 48 states, 2304 transitions, training time: 3.2s"
10. Administrator klika przycisk „Train Naive Bayes Model"
11. System wywołuje endpoint `POST /api/admin/probabilistic/train-bayes`
12. Backend uruchamia `NaiveBayesEngine.train()`:
    - Pobiera wszystkich użytkowników z zamówieniami
    - Dla każdego użytkownika ekstraktuje 5 cech: liczba zamówień, średnia wartość, dni od ostatniego, ulubiona kategoria, częstotliwość
    - Tworzy etykiety binarne (aktywny/nieaktywny) na podstawie heurystyki (zakup w ostatnich 30 dniach)
    - Oblicza prawdopodobieństwa warunkowe dla każdej cechy: $P(feature|class)$
    - Oblicza prawdopodobieństwa a priori: $P(active)$, $P(inactive)$
    - Waliduje model na zbiorze testowym (80%/20% split)
    - Zapisuje model w tabeli cache
13. System wyświetla komunikat: „Naive Bayes trained successfully. Training samples: 200, Test accuracy: 78%, training time: 1.8s"
14. Administrator widzi zaktualizowane timestampy i metryki

Warunki końcowe: Modele probabilistyczne (Markov Chain i Naive Bayes) zostały wytrenowane na aktualnych danych, rekomendacje są aktualne

Scenariusz 22: Debugowanie modeli probabilistycznych

Aktor: Administrator
Warunki początkowe: Administrator jest zalogowany, modele zostały wytrenowane
Przebieg scenariusza głównego - Markov Chain:

1. Administrator otwiera panel administracyjny
2. Administrator klika sekcję „Debug"
3. Administrator wybiera zakładkę „Markov Chain Debug"
4. System wyświetla panel debugowania zawierający:
   - Wzór matematyczny macierzy przejść z wygładzaniem Laplace'a
   - Statystyki: liczba stanów (48 kategorii), łączna liczba przejść, wygładzanie $\alpha = 1.0$, timestamp ostatniego treningu
   - Wizualizację macierzy przejść jako heatmap 48x48
5. Administrator wybiera kategorię źródłową z listy rozwijanej (np. „Laptopy")
6. System wyświetla Top 10 kategorii docelowych z najwyższymi prawdopodobieństwami przejścia:
   - Akcesoria komputerowe: 0.45 (45%)
   - Peryferia: 0.30 (30%)
   - Torby i plecaki: 0.15 (15%)
   - Oprogramowanie: 0.05 (5%)
   - Inne kategorie: $<$0.05 (łącznie 5%)
7. Administrator klika „Show Transition Examples"
8. System pobiera przykładowe rzeczywiste sekwencje zamówień użytkowników, które wygenerowały te przejścia
9. Administrator weryfikuje sensowność przejść
10. Administrator przełącza zakładkę na „Naive Bayes Debug"
11. System wyświetla panel Naive Bayes zawierający:
    - Wzór twierdzenia Bayesa
    - Statystyki: liczba próbek treningowych, accuracy, precision, recall, F1-score
    - Tabelę prawdopodobieństw warunkowych $P(feature|class)$ dla każdej z 5 cech
    - Prawdopodobieństwa a priori: $P(active) = 0.65$, $P(inactive) = 0.35$
12. Administrator wprowadza ID użytkownika w pole „User ID for prediction"

13. System pobiera profil użytkownika i wykonuje predykcję krok po kroku:
    - Ekstraktuje cechy użytkownika
    - Pokazuje obliczenia $P(active|features)$ z rozpisaniem każdego czynnika
    - Wyświetla końcowe prawdopodobieństwo oraz klasyfikację (active/inactive)
14. Administrator weryfikuje poprawność predykcji

Warunki końcowe: Administrator zweryfikował poprawność działania modeli probabilistycznych (Markov Chain i Naive Bayes)