

Data 26:04:2024

Rok akademicki 2023/24

Moduł sumatywny: *testLearning*

Dokumentacja

Kacper Czernik 242371

Mateusz Grzeszczak 242398

Kacper Jagodziński 242403

Jakub Szymczak 242548

Wprowadzenie

Niniejszy dokument ma na celu przedstawić skrócony opis projektu *testLearning*, czyli aplikacji przeznaczonej na komputery osobiste, której celem jest ułatwienie nauki testowania oraz projektowania testów jednostkowych również tych przeznaczonych do testowania interfejsu użytkownika.

Opis Problemu

Testowanie aplikacji jest ważnym elementem programowania, mimo tego jest ono często pomijane w trakcie nauki bądź też nacisk na testowanie aplikacji nie jest wystarczający aby zapewnić zrozumienie tego niejednokrotnie skomplikowanego procesu, jest to szczególnie widoczne u osób uczących się programować na własną rękę przez co wielu z nich poszukuje możliwości nauczenia się testowania aplikacji bądź też przetestowania swoich umiejętności na realnym problemie. Nasz projekt ma na celu zapewnić zainteresowanym osobą taką możliwość wraz z dostarczeniem materiałów pozwalających na ocenę efektów własnej pracy.

Potencjalni użytkownicy

Nasz projekt tworzony jest z myślą o indywidulanych użytkownikach uczących się od podstaw bądź chcących doskonalić swoje umiejętności testowania aplikacji, uwzględniając przy tym testy interfejsu użytkownika. W dalszej części dokumentu użytkownicy nazywani będą użytkownikami końcowymi, bądź testującymi.

Założenia i ograniczenia

Projekt ma na celu rozwiązanie przedstawione problemu, uwzględnia on utworzenie dwóch osobnych wersji tej samej aplikacji, będącej prostym sklepem internetowym wraz ze wszystkimi podstawowymi funkcjonalnościami oraz uproszczoną wersją bazy danych.

Pierwszą wersją będzie tak zwana wersja referencyjna ma być bezbłędną implementacją sklepu internetowego z uwzględnieniem utworzonego przez nas zestaw wyczerpujących testów, zarówno tych jednostkowych jak i integracyjnych, pozwalających stwierdzić bezbłądność aplikacji oraz stanowiący przykładowe podejście do problemu testowania aplikacji. Co umożliwi uczącym się na porównanie swojego podejścia jak i ocenę skuteczności wprowadzonych rozwiązań.

Druga wersja będzie składać się tylko z samego sklepu zawierającego celowo umieszczone przez nas błędy w kodzie, tu należy wspomnieć że błędy dotyczyć będą zarówno samej logiki sklepu jak i jego interfejsu. Wykrycie wspomniane błędów poprzez napisanie odpowiednich testów będzie stanowić wyzwanie dla użytkowników końcowych, czego skutkiem będzie zapoznanie się z problemem kompleksowego testowania aplikacji webowych.

Projekt sklepu internetowego uwzględnia istnienie uproszczonej wersji bazy danych, wraz z jej wstępnym wypełnieniem przykładowymi danymi. Ważnym elementem będzie również dostarczenie możliwości szybkiego powrotu do pierwotnej wersji bazy (tej przed wprowadzonymi przez użytkowników zmianami).

Połączenie wersji referencyjnej oraz wersji stanowiącej swojego rodzaju „piaskownicę” pozwoli użytkownikowi na indywidualną naukę oraz wykształcenie własnego podejścia do zagadnienia testowania aplikacji, a dostarczona przez nas do projektu dokumentacja pozwoli na badanie indywidualnych postępów oraz sprawdzanie szczelności utworzonych testów.

Zaplanowane błędy

Poniżej opisane zostały przygotowane przez nas propozycję błędów celowo wprowadzanych w wersji piaskownicy naszej aplikacji. Błędy te mają na celu stanowić wyzwania, które użytkownicy mogą napotkać podczas pracy z testowaniem wszelakich aplikacji, symulując realistyczne scenariusze, z jakimi mogą się spotkać testerzy oprogramowania:

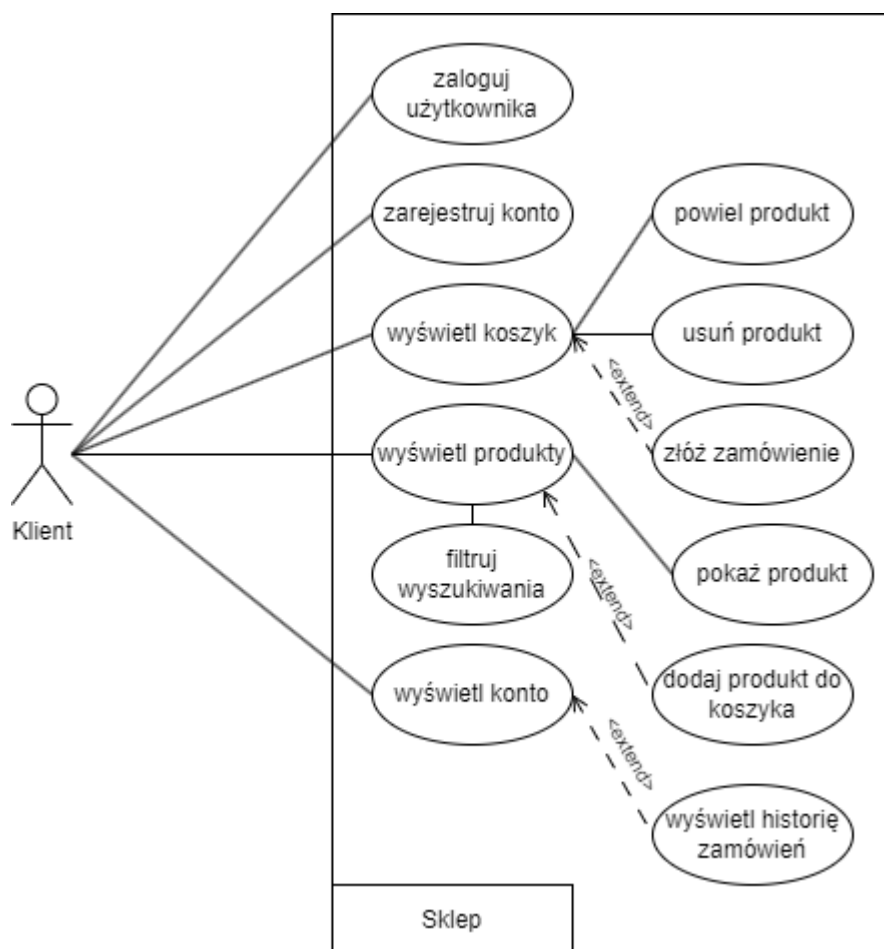
- **Błąd w implementacji endpointu** - W tej sytuacji, błąd w implementacji konkretnego endpointu API prowadzi do przerwania komunikacji między podstronami aplikacji. Na przykład, nieprawidłowo zdefiniowany endpoint odpowiedzialny za załadowanie opisu produktów może powodować, że sekcja opisu na stronie produktu nie zostanie wcale wyświetlona. Taki błąd może wynikać z niepoprawnego adresu URL endpointu, błędów w logice obsługi żądań lub problemów z serializacją i deserializacją danych.
- **Błąd połączenia UI z back-endem przy dodawaniu produktu do koszyka** - Tu występuje błąd w interfejsie użytkownika (UI), gdzie po naciśnięciu przycisku "Dodaj do koszyka" produkt faktycznie nie zostaje dodany do koszyka zakupowego. Problem ten może być spowodowany nieprawidłowym połączeniem między front-endem a back-endem, np. przez niepoprawne wywołanie API lub błędy w obsłudze odpowiedzi od serwera. Użytkownik może być wprowadzony w błąd, ponieważ interfejs może wydawać się responsywny, ale bez widocznych skutków działania.
- **Błąd pobierania z tabeli UserOrders** - Ten błąd dotyczy problemów z działaniem bazy danych, szczególnie przy pobieraniu danych z tabeli przechowującej informacje o zamówieniach użytkownika. Może to skutkować błędnym lub pustym wyświetleniem zawartości koszyka, błędami w historii zakupów lub nieprawidłowym wyświetlaniem szczegółów zamówienia. Problem może wynikać z błędów w zapytaniach SQL, uszkodzeniem struktury bazy danych lub problemami z synchronizacją danych.
- **Problem z zapisem rejestracji użytkownika do bazy danych** - Tutaj mamy do czynienia z błędem w procesie rejestracji nowych użytkowników, gdzie pomimo pozornie pomyślnego przejścia przez proces rejestracji, dane użytkownika nie są zapisywane w bazie danych. Oznacza to, że podczas próby logowania, system nie może znaleźć danych nowego użytkownika, uniemożliwiając mu dostęp do aplikacji. Błąd ten może być spowodowany nieprawidłową konfiguracją bazy danych, błędami w kodzie odpowiadającym za zapis danych lub problemami z transakcjami bazodanowymi.

Diagramy dla aplikacji sklepu

Poniżej zaprezentowane zostały przygotowane przez nas diagramy dotyczące aplikacji sklepu internetowego. Diagramy te odnoszą się zarówno do wersji referencyjnej jak i wersji piaskownicy i mają za zadanie jedynie przedstawić główne założenia aplikacji.

Diagram przypadków użycia

grafika 1: diagram przypadków użycia



Scenariusze przypadków użycia

W poniższych scenariuszach użytkownik oznacza osobę uczącą się od podstaw lub chcącą doskonalić swoje umiejętności testowania aplikacji.

Scenariusz 1: Zarejestruj konto

Identyfikator	1101
Nazwa przypadku użycia	Zarejestruj konto
Poziom ważności	Średni
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia nowemu użytkownikowi stworzenie konta w systemie, co jest niezbędne do dostępu do funkcji wymagających autoryzacji, takich jak składanie zamówień, dostęp do historii zamówień, oraz uczestnictwo w testowaniu aplikacji.
Warunki wstępne	Użytkownik nie jest zalogowany ani nie posiada konta w systemie.
Warunki końcowe	Użytkownik ma stworzone konto i możliwość zalogowania się do systemu.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Zarejestruj się". 2. System prosi o podanie niezbędnych informacji rejestracyjnych, takich jak nazwa użytkownika, adres e-mail, hasło i ewentualnie inne dane jak numer telefonu czy wiek. 3. Użytkownik wypełnia formularz i zatwierdza rejestrację. 4. System weryfikuje wprowadzone dane, sprawdzając ich poprawność i unikalność nazwy użytkownika oraz adresu e-mail. 5. System tworzy nowe konto i informuje użytkownika o pomyślnej rejestracji. 6. Użytkownik ma możliwość bezpośredniego zalogowania się do systemu lub przejścia na stronę główną.
Alternatywne przepływy wydarzeń	<p>A1. Błędne lub niekompletne dane rejestracyjne:</p> <p>6.1. Jeśli którykolwiek z wymaganych danych jest niepoprawny lub brakuje, system wyświetla stosowny komunikat o błędzie i prosi użytkownika o poprawienie danych.</p> <p>6.2. Użytkownik koryguje dane i ponownie zatwierdza formularz.</p> <p>A2. Nazwa użytkownika lub adres e-mail już istnieje:</p> <p>6.1 Jeśli wybrana nazwa użytkownika lub adres e-mail są już używane, system informuje o tym fakcie i prosi o wybranie innej nazwy użytkownika lub podanie innego adresu e-mail.</p> <p>6.2. Użytkownik wprowadza nowe dane i zatwierdza rejestrację.</p>

Scenariusz 2: Zaloguj użytkownika

Identyfikator	1102
Nazwa przypadku użycia	Zaloguj użytkownika
Poziom ważności	Średni
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikowi zalogowanie się do systemu w celu uzyskania dostępu do funkcji zarezerwowanych dla zalogowanych użytkowników, takich jak obsługa koszyka, składanie zamówień, dostęp do historii zamówień, itp.
Warunki wstępne	Użytkownik musi być zarejestrowany w systemie.
Warunki końcowe	Użytkownik jest zalogowany i ma dostęp do dodatkowych funkcji sklepu.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Zaloguj się". 2. System prosi o podanie nazwy użytkownika (lub adresu e-mail) i hasła. 3. Użytkownik wprowadza swoje dane i zatwierdza. 4. System weryfikuje dane i autoryzuje użytkownika. 5. Użytkownik zostaje przekierowany na stronę główną lub na poprzednio przeglądany stronę z dostępem do dodatkowych funkcji.
Alternatywne przepływy wydarzeń	<p>A1. Niepoprawne dane logowania:</p> <p>5.1. Jeśli nazwa użytkownika lub hasło są niepoprawne, system informuje o błędzie i prosi o ponowne wprowadzenie danych.</p> <p>5.2. Użytkownik może spróbować ponownie lub wybrać opcję odzyskiwania hasła.</p> <p>A2. Zapomniane hasło:</p> <p>5.1. Jeśli użytkownik zapomni hasła, może wybrać opcję "Zapomniałem hasła".</p> <p>5.2. System prosi o podanie adresu e-mail powiązanego z kontem.</p> <p>5.3. Użytkownik wprowadza adres e-mail i zatwierdza.</p> <p>5.4. System wysyła link do zresetowania hasła na podany adres e-mail.</p> <p>5.5. Użytkownik postępuje zgodnie z instrukcjami w e-mailu, aby ustawić nowe hasło.</p>

Identyfikator	1103
Nazwa przypadku użycia	Wyświetl produkty
Poziom ważności	Ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom przeglądanie listy dostępnych produktów w sklepie internetowym, wraz z podstawowymi informacjami na ich temat.
Warunki wstępne	Brak (Funkcjonalność dostępna dla wszystkich odwiedzających sklep).
Warunki końcowe	Użytkownik przegląda listę produktów.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wchodzi na stronę główną sklepu internetowego. 2. System wyświetla listę dostępnych produktów, wraz z ich nazwami, cenami oraz miniaturkami zdjęć. 3. Użytkownik może przewijać listę, aby zobaczyć więcej produktów. 4. Jeśli użytkownik chce uzyskać więcej informacji o produkcie, może kliknąć na wybrany produkt, aby przejść do jego szczegółowego widoku (co jest tematem innego przypadku użycia).
Alternatywne przepływy wydarzeń	<p>A1. Brak produktów do wyświetlenia:</p> <p>4.1. Jeśli w sklepie nie ma dostępnych produktów, system wyświetla stosowny komunikat informujący o braku produktów.</p> <p>A2. Błąd ładowania produktów:</p> <p>4.1 Jeśli wystąpi problem z ładowaniem listy produktów (np. problem z połączeniem z bazą danych), system wyświetla komunikat o błędzie i prosi użytkownika o odświeżenie strony.</p>

Scenariusz 4: Pokaż produkt

Identyfikator	1104
Nazwa przypadku użycia	Pokaż produkt
Poziom ważności	Ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom wyświetlenie szczegółowych informacji o produkcie, w tym opisu, ceny oraz dostępnych opcji, takich jak rozmiar czy kolor (jeśli dotyczy).
Warunki wstępne	Użytkownik znajduje się na liście produktów i wybiera produkt, który chce bliżej zobaczyć.
Warunki końcowe	Użytkownik przegląda szczegółowe informacje o produkcie.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera produkt z listy produktów, klikając na jego nazwę lub obrazek. 2. System wyświetla stronę produktu z pełnym opisem, większymi zdjęciami, ceną produktu oraz dostępnymi opcjami, jak wybór rozmiaru czy koloru (jeśli dotyczy). 3. Użytkownik może przeczytać opis produktu, przejrzeć galerię zdjęć i zapoznać się z dostępnymi opcjami. 4. Użytkownik może wybrać ilość produktów oraz inne dostępne opcje (rozmiar, kolor) i dodać produkt do koszyka lub powrócić do listy produktów.
Alternatywne przepływy wydarzeń	<p>A1. Produkt niedostępny:</p> <p>4.1. Jeśli wybrany produkt jest tymczasowo niedostępny, system wyświetla komunikat o braku dostępności oraz przewidywanym czasie uzupełnienia zapasów.</p> <p>A2. Błąd ładowania szczegółów produktu:</p> <p>4.1. W przypadku problemów z ładowaniem szczegółów produktu (np. błąd serwera), system informuje użytkownika o błędzie i proponuje odświeżenie strony lub powrót do listy produktów.</p>

Scenariusz 5: Dodaj produkt do koszyka

Identyfikator	1105
Nazwa przypadku użycia	Dodaj produkt do koszyka
Poziom ważności	Średni
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom dodanie wybranego produktu do koszyka zakupowego wraz z wybraną ilością i innymi dostępnymi opcjami produktu.
Warunki wstępne	Użytkownik znajduje się na stronie szczegółowych informacji o produkcie.
Warunki końcowe	Wybrany produkt zostaje dodany do koszyka zakupowego użytkownika.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera żądaną ilość produktu oraz inne dostępne opcje, takie jak rozmiar lub kolor (jeśli dotyczy). 2. Użytkownik klika przycisk "Dodaj do koszyka" na stronie szczegółów produktu. 3. System weryfikuje dostępność wybranego produktu oraz wybrane opcje. 4. System dodaje produkt do koszyka zakupowego, aktualizując informacje o ilości produktów i łącznej kwocie. 5. System wyświetla potwierdzenie, że produkt został dodany do koszyka, i oferuje opcje przejścia do koszyka lub kontynuowania zakupów.
Alternatywne przepływy wydarzeń	<p>A1. Produkt niedostępny:</p> <p>5.1. Jeśli wybrana ilość produktu przekracza dostępny zapas, system informuje użytkownika o niedostępności i prosi o zmniejszenie ilości.</p> <p>A2. Błąd systemu:</p> <p>5.1. W przypadku wystąpienia błędu systemowego uniemożliwiającego dodanie produktu do koszyka (np. błąd komunikacji z bazą danych), system informuje użytkownika o problemie i sugeruje ponowną próbę później.</p>

Scenariusz 6: Filtruj wyszukiwania

Identyfikator	1106
Nazwa przypadku użycia	Filtruj wyszukiwania
Poziom ważności	Mało ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom stosowanie filtrów do listy produktów, aby zawęzić wyniki wyszukiwania do tych, które najbardziej odpowiadają ich preferencjom.
Warunki wstępne	Użytkownik ma dostęp do listy produktów w sklepie internetowym.
Warunki końcowe	Lista produktów jest filtrowana zgodnie z zastosowanymi kryteriami użytkownika.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Filtruj", dostępną na stronie z listą produktów. 2. System prezentuje dostępne filtry, takie jak kategoria, zakres cen, marka, ocena itp. 3. Użytkownik wybiera jeden lub więcej filtrów, określając swoje preferencje. 4. Użytkownik zatwierdza wybrane filtry, klikając na przycisk "Zastosuj filtry" lub podobny. 5. System przetwarza wybrane filtry i aktualizuje listę produktów, wyświetlając tylko te, które spełniają określone kryteria. 6. Użytkownik przegląda filtrowane wyniki wyszukiwania.
Alternatywne przepływy wydarzeń	<p>A1. Brak produktów spełniających kryteria:</p> <p>6.1. Jeśli żaden produkt nie spełnia wybranych kryteriów, system wyświetla komunikat informujący o braku wyników i sugeruje złagodzenie kryteriów filtrowania.</p> <p>A2. Zmiana kryteriów filtrowania:</p> <p>6.1. Użytkownik może w każdej chwili zmodyfikować wybrane filtry, dodając nowe kryteria lub usuwając istniejące, co skutkuje ponownym przefiltrowaniem listy produktów.</p> <p>A3. Usunięcie wszystkich filtrów:</p> <p>6.1. Użytkownik ma możliwość usunięcia wszystkich zastosowanych filtrów, co przywraca wyświetlanie pełnej listy produktów.</p>

Identyfikator	1107
Nazwa przypadku użycia	Wyświetl koszyk
Poziom ważności	Średni
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom przeglądanie zawartości ich koszyka zakupowego, w tym produktów, które dodali, wraz z ilościami i cenami.
Warunki wstępne	Użytkownik dodał co najmniej jeden produkt do koszyka.
Warunki końcowe	Użytkownik przegląda zawartość swojego koszyka zakupowego.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Koszyk" lub ikonę koszyka zakupowego dostępną na stronie sklepu. 2. System wyświetla stronę koszyka zakupowego, zawierającą listę wszystkich produktów dodanych do koszyka, wraz z ich ilościami, cenami jednostkowymi oraz zdjęciem produktu. 3. Dla każdego produktu w koszyku, użytkownik może zmienić ilość sztuk lub usunąć produkt z koszyka. 4. Na dole strony koszyka wyświetlana jest łączna kwota do zapłaty za wszystkie produkty, uwzględniając wybrane ilości. 5. Użytkownik ma możliwość kontynuowania zakupów (powrót do przeglądania produktów) lub przejścia do realizacji zamówienia (przycisk "Przejdź do kasy" lub podobny).
Alternatywne przepływy wydarzeń	<p>A1. Koszyk jest pusty:</p> <p>5.1. Jeśli koszyk nie zawiera żadnych produktów, system wyświetla komunikat informujący o pustym koszyku oraz zapraszający do kontynuowania zakupów.</p> <p>A2. Zmiana ilości produktu:</p> <p>5.1. Użytkownik może zmienić ilość konkretnego produktu w koszyku, co skutkuje automatyczną aktualizacją łącznej kwoty do zapłaty.</p> <p>A3. Usunięcie produktu z koszyka:</p> <p>5.1. Użytkownik ma możliwość usunięcia produktu z koszyka, co powoduje jego natychmiastowe wycofanie z listy i aktualizację łącznej kwoty zamówienia.</p>

Identyfikator	1108
Nazwa przypadku użycia	Powiel produkt
Poziom ważności	Mało ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom powielenie wybranego produktu w koszyku zakupowym, umożliwiając zakup tego samego produktu z różnymi opcjami bez konieczności ponownego wyszukiwania produktu w sklepie.
Warunki wstępne	Użytkownik ma co najmniej jeden produkt w koszyku zakupowym.
Warunki końcowe	Kopia produktu zostaje dodana do koszyka, umożliwiając użytkownikowi dostosowanie opcji dla powielonego produktu.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wchodzi na stronę koszyka zakupowego, aby przeglądać dodane produkty. 2. Użytkownik wybiera produkt, który chce powielić, klikając przycisk "Powiel" lub ikonę obok produktu w koszyku. 3. System tworzy kopię wybranego produktu w koszyku z tą samą ilością i opcjami, co oryginalny produkt. 4. Użytkownik może następnie dostosować opcje dla powielonego produktu, np. zmienić rozmiar lub kolor. 5. Zaktualizowany koszyk jest wyświetlany użytkownikowi, zawierając oryginalny produkt oraz jego kopię z ewentualnymi zmianami.
Alternatywne przepływy wydarzeń	<p>A1. Brak opcji powielenia dla niektórych produktów:</p> <p>5.1. Jeśli dany produkt nie kwalifikuje się do powielenia (np. z powodu ograniczeń magazynowych), system nie wyświetla opcji "Powiel" dla tego produktu.</p> <p>A2. Ograniczenie ilości kopii:</p> <p>5.1. System może nałożyć ograniczenia na ilość kopii danego produktu, które można dodać do koszyka, aby zapobiec nadmiernemu zamawianiu.</p>

Scenariusz 9: Usuń produkt

Identyfikator	1109
Nazwa przypadku użycia	Usuń produkt
Poziom ważności	Średni
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom usunięcie produktu z ich koszyka zakupowego.
Warunki wstępne	Użytkownik ma co najmniej jeden produkt w koszyku zakupowym.
Warunki końcowe	Wybrany produkt zostaje usunięty z koszyka zakupowego użytkownika.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik otwiera koszyk zakupowy, aby przeglądać dodane produkty. 2. Obok każdego produktu w koszyku znajduje się przycisk "Usuń" lub ikona kosza. 3. Użytkownik klika przycisk "Usuń" obok produktu, który chce usunąć z koszyka. 4. System pyta o potwierdzenie usunięcia produktu, aby zapobiec przypadkowemu usunięciu. 5. Po potwierdzeniu przez użytkownika, system usuwa wybrany produkt z koszyka. 6. System aktualizuje zawartość koszyka i łączną kwotę do zapłaty, wykluczając usunięty produkt.
Alternatywne przepływy wydarzeń	<p>A1. Anulowanie usunięcia produktu:</p> <p>5.1. Jeśli użytkownik zdecyduje się nie usuwać produktu, może anulować operację w oknie potwierdzenia.</p> <p>A2. Usunięcie wszystkich produktów:</p> <p>2.1. Użytkownik może wybrać opcję "Usuń wszystkie", aby usunąć wszystkie produkty z koszyka jednocześnie.</p>

Identyfikator	1110
Nazwa przypadku użycia	Złóż zamówienie
Poziom ważności	Ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom finalizację procesu zakupowego poprzez przeglądanie podsumowania zamówienia, wprowadzenie danych do wysyłki, wybór metody płatności i potwierdzenie zakupu.
Warunki wstępne	Użytkownik ma co najmniej jeden produkt w koszyku i jest gotowy do finalizacji zakupu.
Warunki końcowe	Zamówienie zostało złożone, a użytkownik otrzymuje potwierdzenie zakupu.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Przejdź do kasy" z poziomu koszyka zakupowego. 2. System wyświetla podsumowanie zamówienia, w tym listę produktów, ich ilość, cenę, oraz łączną kwotę do zapłaty. 3. Użytkownik wprowadza lub potwierdza dane do wysyłki, w tym adres dostawy. 4. Użytkownik wybiera preferowaną metodę płatności (np. karta kredytowa, przelew bankowy, płatność przy odbiorze). 5. System prosi o potwierdzenie szczegółów zamówienia i akceptację warunków sklepu. 6. Użytkownik potwierdza zamówienie, klikając przycisk "Złóż zamówienie" lub podobny. 7. System przetwarza zamówienie, realizuje płatność (w zależności od wybranej metody) i wyświetla potwierdzenie złożenia zamówienia wraz z numerem zamówienia i szacowanym czasem dostawy.
Alternatywne przepływy wydarzeń	<p>A1. Błąd w danych do wysyłki lub płatności:</p> <p>7.1. Jeśli system wykryje błąd lub brakujące informacje w danych do wysyłki lub płatności, wyświetla stosowny komunikat błędu i prosi użytkownika o korektę.</p> <p>A2. Anulowanie przed finalizacją:</p> <p>Użytkownik może w dowolnym momencie anulować proces składania zamówienia, wracając do koszyka lub kontynuując zakupy.</p> <p>A3. Problemy z płatnością:</p> <p>7.1. W przypadku problemów z autoryzacją płatności, system informuje użytkownika o niepowodzeniu i prosi o wybranie innej metody płatności lub ponowne wprowadzenie danych.</p>

Scenariusz 11: Wyświetl konto

Identyfikator	1111
Nazwa przypadku użycia	Wyświetl konto
Poziom ważności	Mało ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Umożliwia użytkownikom przeglądanie informacji o swoim koncie, w tym danych osobowych, historii zamówień, ustawień konta i preferencji.
Warunki wstępne	Użytkownik jest zalogowany do swojego konta.
Warunki końcowe	Użytkownik przegląda informacje związane ze swoim kontem.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Moje konto" lub podobną z menu użytkownika. 2. System wyświetla panel zarządzania kontem, w którym dostępne są różne sekcje, takie jak: dane osobowe, ustawienia konta, preferencje. 3. Użytkownik może przeglądać swoje dane osobowe, w tym adres do wysyłki, adres e-mail i numer telefonu. 4. W sekcji preferencji, użytkownik może przeglądać zarządzać swoimi preferencjami dotyczącymi komunikacji marketingowej, zainteresowań produktowych itp.
Alternatywne przepływy wydarzeń	Brak.

Scenariusz 12: Zmodyfikuj dane konta

Identyfikator	1112
Nazwa przypadku użycia	Zmodyfikuj dane konta
Poziom ważności	Mało ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Pozwala użytkownikom na edycję i aktualizację danych swojego konta w systemie.
Warunki wstępne	Użytkownik jest zalogowany i znajduje się w panelu zarządzania kontem.
Warunki końcowe	Dane konta użytkownika zostały zaktualizowane.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wchodzi w sekcję "Ustawienia konta" lub podobną w panelu zarządzania kontem. 2. System wyświetla formularz z aktualnymi danymi użytkownika, w tym adresem e-mail, hasłem, informacjami o adresie wysyłki itd. 3. Użytkownik dokonuje pożądanych zmian w swoich danych, takich jak zmiana adresu e-mail, hasła, dodanie lub edycja adresów wysyłki. 4. Po wprowadzeniu zmian, użytkownik klika przycisk "Zapisz zmiany" lub podobny. 5. System weryfikuje wprowadzone zmiany (np. poprawność adresu e-mail, siłę hasła) i prosi o potwierdzenie, jeśli jest to wymagane (np. potwierdzenie przez e-mail w przypadku zmiany adresu e-mail). 6. Po zatwierdzeniu zmian, system aktualizuje dane konta i wyświetla komunikat o pomyślnym zaktualizowaniu danych.
Alternatywne przepływy wydarzeń	<p>A1. Błędy w formularzu:</p> <p>6.1. Jeśli użytkownik wprowadzi nieprawidłowe dane (np. błędny format adresu e-mail), system wyświetla odpowiednie komunikaty błędów i prosi o korektę.</p> <p>A2. Anulowanie zmian:</p> <p>Użytkownik ma możliwość anulowania wprowadzonych zmian przed ich zapisaniem, co przywraca poprzednie dane.</p> <p>A3. Wymagane potwierdzenie zmian:</p> <p>6.1. Dla niektórych zmian, takich jak zmiana hasła czy adresu e-mail, system może wymagać dodatkowego potwierdzenia, np. poprzez kliknięcie przycisku potwierdź.</p>

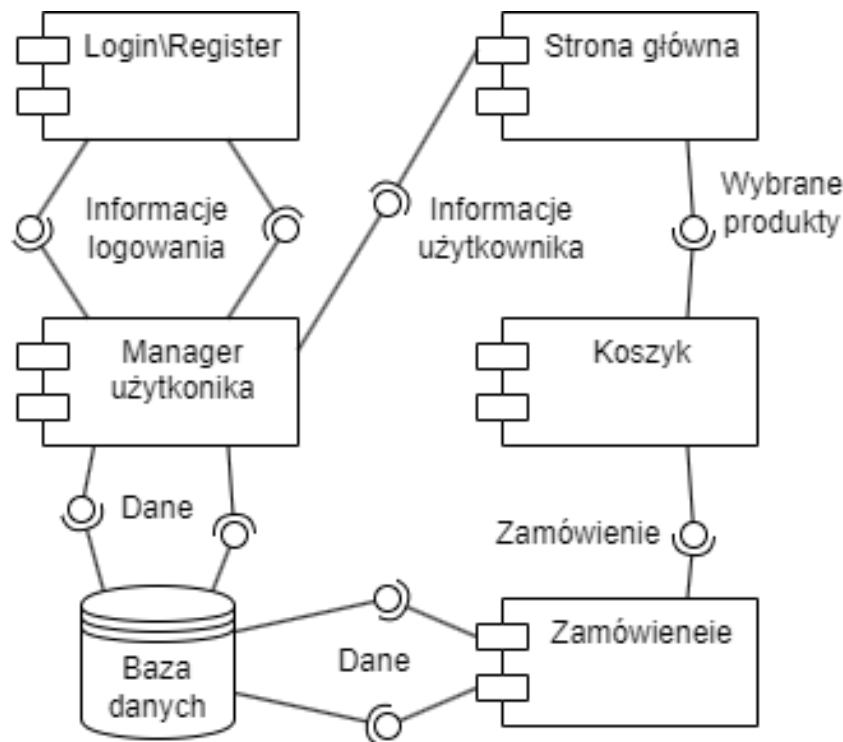
Scenariusz 13: Wyświetl historię zamówień

Identyfikator	1113
Nazwa przypadku użycia	Wyświetl historię zamówień
Poziom ważności	Mało ważny
Typ przypadku użycia	Podstawowy
Aktorzy	Użytkownik
Krótki opis	Pozwala użytkownikom na przeglądanie ich wcześniejszych zakupów, w tym dat zamówień, statusów zamówień, szczegółów produktów i kosztów.
Warunki wstępne	Użytkownik jest zalogowany do swojego konta.
Warunki końcowe	Użytkownik przegląda historię swoich zamówień.
Główny przepływ wydarzeń	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję "Historia zamówień" lub podobną z menu konta użytkownika. 2. System wyświetla listę wszystkich zamówień złożonych przez użytkownika, wraz z podstawowymi informacjami takimi jak numer zamówienia, data zamówienia, status zamówienia i całkowita kwota. 3. Użytkownik może kliknąć na poszczególne zamówienie, aby zobaczyć bardziej szczegółowe informacje, w tym listę zakupionych produktów, ich ilość, cenę jednostkową oraz koszty wysyłki.
Alternatywne przepływy wydarzeń	<p>A1. Brak historii zamówień:</p> <p>3.1. Jeśli użytkownik nie dokonał jeszcze żadnych zakupów, system wyświetla komunikat o braku historii zamówień.</p>

Diagram komponentów

Omawiając diagram komponentów należy pamiętać że celem projektu jest dostarczenie zaledwie uproszczonej wersji sklepu internetowego, dlatego też zastosowane zostały pewne uproszczenia, najważniejszym z nich jest fakt że zawartość koszyka przechowywana jest lokalnie, a nie tak jak jest w standardzie w bazie danych.

grafika 2: diagram komponentów



1. **Login/Register (Logowanie/Rejestracja):**
 - Ten komponent odpowiada za proces logowania i rejestracji użytkowników w aplikacji. Zapewnia interfejs użytkownika umożliwiający użytkownikom zalogowanie się do konta lub utworzenie nowego konta w systemie.
2. **Manager użytkownika (User Manager):**
 - Komponent zarządzający użytkownikami w systemie. Odpowiada za obsługę operacji związanych z użytkownikami, takich jak dodawanie, usuwanie, edycja profili użytkowników oraz zarządzanie ich uprawnieniami.
3. **Baza Danych (Database):**
 - Komponent przechowujący dane użytkowników, informacje o produktach, zamówieniach itp. Odpowiada za przechowywanie i zarządzanie danymi w aplikacji.
4. **Strona główna (Homepage):**
 - Ten komponent to główny interfejs użytkownika, który jest pierwszym miejscem, które użytkownicy widzą po zalogowaniu się do aplikacji lub odwiedzeniu strony internetowej. Strona główna może zawierać różne sekcje, takie jak promowane produkty, oferty specjalne, kategorie produktów itp.

5. **Koszyk (Shopping Cart):**

- Komponent odpowiedzialny za obsługę koszyka zakupowego użytkowników. Pozwala użytkownikom dodawać produkty do koszyka, usuwać produkty z koszyka, aktualizować ilość produktów itp.

6. **Zamówienie (Order):**

- Komponent zarządzający procesem składania zamówień w aplikacji. Odpowiada za zbieranie informacji o produktach, adresie dostawy, opcjach płatności itp. oraz generowanie zamówienia do realizacji.

Analiza istniejących rozwiązań

Istnieje wiele platform oferujących bogaty wybór kursów bądź testów związanych z testowaniem oprogramowania, które pozwalają zarówno na zdobycie podstawowej wiedzy, jak i pogłębianie umiejętności w tej dziedzinie. Niewiele z nich udostępnia jednak materiały pozwalające na własnoręczne doskonalenie zdobytych umiejętności bądź nie prowadzi nauki po z góry wytyczonych ścieżkach niejednokrotnie nie mających większego związku z realnie występującymi problemami. Analizując dostępne materiały możemy również zauważyć że duża część z nich stawia nacisk na TDD (*test driven development*).

Poniżej zaprezentowane zostały utworzone przez nas krótkie opisy trzech popularnych platform umożliwiających naukę testowania oprogramowania:

1. **Udemy:** *Udemy* to platforma e-learningowa oferująca kursy online na różnorodne tematy, w tym testowanie oprogramowania. *Udemy* gromadzi szeroki wybór kursów prowadzonych przez doświadczonych specjalistów z branży, zarówno dla początkujących, jak i zaawansowanych uczestników. Kursy skupione są na praktycznych zadaniach oraz projektach.
2. **Coursera:** *Coursera* to platforma edukacyjna współpracująca z uniwersytetami i instytucjami na całym świecie. Oferuje kursy z różnych dziedzin, w tym z testowania oprogramowania. Kursy na Courserze są często zbudowane wokół programów akademickich i zawierają materiały wideo, ćwiczenia oraz różnorodne testy.
3. **Test Automation University (TAU):** TAU to darmowa platforma edukacyjna skupiająca się na automatyzacji testów oprogramowania. Stworzona przez *Testim.io*, oferuje szereg kursów prowadzonych przez ekspertów z branży. Kursy na TAU pokrywają różne aspekty testowania automatycznego i są aktualizowane regularnie.

Tabela 1: porównanie istniejących rozwiązań

	<i>testLearning</i>	<i>Udemy</i>	<i>Coursera</i>	<i>Test Automation University</i>
Metodologia nauki	Proponuje unikalne podejście poprzez połączenie wersji referencyjnej sklepu internetowego z wersją zawierającą błędy. To podejście skupia się na praktycznej nauce przez doświadczenie, gdzie użytkownik nie tylko uczy się na przykładach idealnych rozwiązań, ale również staje przed wyzwaniem naprawy rzeczywistych błędów w kodzie.	Oferują kursy oparte na wideo-lekcjach z praktycznymi zadaniami. Metodologia nauki jest bardziej tradycyjna i skupia się na przekazywaniu wiedzy przez ekspertów, co może być mniej angażujące niż rozwiązanie problemów w rzeczywistym projekcie.		Skupia się głównie na automatyzacji testów, oferując kursy prowadzone przez ekspertów. Chociaż jest to cenne źródło wiedzy o automatyzacji, może nie obejmować w pełni ręcznego testowania i debugowania, które są istotne w <i>testLearning</i> .
Zakres materiałów	Koncentruje się na testowaniu aplikacji webowych, w szczególności na testowaniu sklepu internetowego, co może być bardziej specyficzne niż oferty innych platform, ale jednocześnie oferuje bezpośrednio zastosowanie w realnych projektach.	Mają szeroki zakres kursów pokrywających różne aspekty testowania oprogramowania, od podstaw po zaawansowane techniki, co sprawia, że są one bardziej wszechstronne.		Skoncentrowany jest wyłącznie na automatyzacji testów, co oznacza, że jego zakres jest bardziej ograniczony niż w przypadku <i>Udemy</i> czy <i>Coursera</i> , ale jednocześnie bardziej skoncentrowany na tej konkretnej dziedzinie.
Interaktywność	Stawia na wysoki poziom interaktywności, umożliwiając użytkownikom eksperymentowanie z kodem, co jest kluczowym aspektem nauki programowania i testowania. Ta praktyczna metoda pozwala na lepsze zrozumienie materiału.	Oferują, również zadania praktyczne, jednak interaktywność może być ograniczona w zależności od kursu.		Oferuje kursy z interaktywnymi elementami, szczególnie w kontekście nauki narzędzi do automatyzacji, co jest zbliżone do podejścia <i>testLearning</i> , chociaż w bardziej niszowej dziedzinie.
Specyfika zadań praktycznych	Wyróżnia się unikalnym podejściem do zadań praktycznych, oferując dwie wersje aplikacji: referencyjną i z błędami do wykrycia i naprawienia. To podejście mocno odzwierciedla rzeczywiste wyzwania, z którymi programiści i testerzy mogą się spotkać w pracy.	zapewniają praktyczne projekty i zadania, które są bardziej zróżnicowane, ale mogą nie zawsze odzwierciedlać realistyczne scenariusze błędów w oprogramowaniu.		koncentruje się na praktycznym zastosowaniu narzędzi do automatyzacji testów, co jest nieco inaczej ukierunkowane niż kompleksowe podejście do testowania oferowane przez <i>testLearning</i> .

Podział funkcjonalności

Poniżej znajdują się przygotowane przez nas wymagania funkcjonalne oraz нефункционалне przeznaczone dla aplikacji sklepu internetowego, zaprezentowane punkty odnoszą się do obu wersji aplikacji (referencyjnej oraz piaskownicy). Funkcjonalności dzielimy na takie dla wersji referencyjnej oraz dla wersji piaskownicy, taki podział jest potrzebny z uwagi na specyfikę naszego projektu. Zagadnienia związane z testowaniem oraz umieszczeniem błędów na „sklepie piaskownicy” zaklasyfikowane zostały do wymagań funkcjonalnych ze względu na naturę tworzonego projektu.

Wymagania Funkcjonalne		
1.	Zarejestruj konto	Identyfikator: 1101
	Aplikacja umożliwia użytkownikom rejestrację nowych w celu uzyskania dostępu do dodatkowych funkcji sklepu.	
2.	Zaloguj użytkownika	Identyfikator: 1102
	Aplikacja umożliwia użytkownikom logowanie się do istniejących kont w celu uzyskania dostępu do dodatkowych funkcji sklepu.	
3.	Wyświetl produkty	Identyfikator: 1103
	Użytkownicy mogą przeglądać dostępne produkty w sklepie.	
4.	Pokaż produkt	Identyfikator: 1104
	Użytkownik może uzyskać szczegółowe informacje o każdym produkcie.	
5.	Dodaj produkt do koszyka	Identyfikator: 1105
	Klienci mogą dodawać wybrane produkty do koszyka jak i manipulować ich ilością w celu zakupowego w celu złożenia zamówienia.	
6.	Filtruj wyszukiwania	Identyfikator: 1106
	Użytkownicy podczas wyszukiwania dostępnych produktów w sklepie może skorzystać z predefiniowanych filtrów wyszukiwania.	
7.	Wyświetl koszyk	Identyfikator: 1107
	Użytkownik może wyświetlić swój koszyk, może przeglądać produkty, które do niego dodał.	
8.	Powiel produkt	Identyfikator: 1108
	Umożliwia użytkownikowi powielenie produktu w swoim koszyku (wraz z jego ilością, licznosc powielonego produktu można potem zmienić).	
9.	Usuń produkt	Identyfikator: 1109
	Umożliwia użytkownikowi usunięcie produktu z koszyka.	
10.	Złóż zamówienie	Identyfikator: 1110
	Umożliwienie zalogowanym użytkownikom złożenia zamówienia na wybrane produkty z koszyka oraz podanie danych niezbędnych do realizacji zamówienia. Złożone zamówienie zostanie odpowiednio zarejestrowane przez system.	

11.	Wyświetl konto	Identyfikator: 1111
	Zalogowaniu użytkownicy zyskają możliwość zobaczenia informacji o swoim koncie.	
12.	Zmodyfikuj dane konta	Identyfikator: 1112
	Zalogowani użytkownicy zyskają możliwość zmiany informacji na swoim koncie.	
13.	Wyświetl historię zamówień	Identyfikator: 1113
	Zalogowani użytkownicy zyskają dostęp do historii swoich zamówień i będą mogli ją wyświetlić.	
14.	Usługa szybkiego resetowania bazy danych	Identyfikator: 1114
	Dodatkowa usługa wykraczająca poza samą aplikację sklepu internetowego, pozwoli na szybkie i wydajne resetowanie bazy danych do jej wersji referencyjnej. Co ma za zadanie ułatwić jej testowanie.	
15.	Dostarczenie testów wersji referencyjnej	Identyfikator: 1301
	Wersja referencyjna aplikacji zawierać będzie kompleksowy zestaw testów (Implementacja testów jednostkowych i integracyjnych), mający na celu zapewnienie bezawaryjności tej wersji aplikacji.	
16.	Wprowadzenie błędów wersji piaskownicy	Identyfikator: 1201
	Wersja sklepu będąca piaskownicą ma zawierać umyślnie wprowadzone błędy pozwalające na naukę testowania aplikacji. Co jest głównym celem naszej aplikacji.	

Tabela 2: Wymagania funkcjonalne

Wymagania Funkcjonalne		
1.	Rejestracja użytkownika	Identyfikator: 1101
	Aplikacja umożliwia użytkownikom rejestrację nowych w celu uzyskania dostępu do dodatkowych funkcji sklepu.	
2.	Logowanie użytkownika	Identyfikator: 1102
	Aplikacja umożliwia użytkownikom logowanie się do istniejących kont w celu uzyskania dostępu do dodatkowych funkcji sklepu.	
3.	Przeglądanie produktów	Identyfikator: 1103
	Użytkownicy mogą przeglądać dostępne produkty w sklepie oraz uzyskać szczegółowe informacje o każdym produkcie.	
4.	Wyszukiwanie produktów	Identyfikator: 1104
	Użytkownicy mogą wyszukiwać dostępne produkty w sklepie oraz korzystać z predefiniowanych filtrów.	
5.	Obsługa koszyka	Identyfikator: 1105
	Klienci mogą dodawać oraz usuwać wybrane produkty do koszyka jak i manipulować ich ilością w celu zakupowego w celu złożenia zamówienia.	

6.	Obsługa zamówień	Identyfikator: 1106
	Umożliwienie zalogowanym użytkownikom złożenia zamówienia na wybrane produkty z koszyka oraz podanie danych niezbędnych do realizacji zamówienia. Złożone zamówienie zostanie odpowiednio zarejestrowane przez system.	
7.	Obsługa historii zamówień	Identyfikator: 1107
	Zalogowani użytkownicy zyskają dostęp do historii swoich zamówień.	
8.	Usługa szybkiego resetowania bazy danych	Identyfikator: 1108
	Dodatkowa usługa wykraczająca poza samą aplikację sklepu internetowego, pozwoli na szybkie i wydajne resetowanie bazy danych do jej wersji referencyjnej. Co ma za zadanie ułatwić jej testowanie.	
9.	Dostarczenie testów wersji referencyjnej	Identyfikator: 1301
	Wersja referencyjna aplikacji zawierać będzie kompleksowy zestaw testów (Implementacja testów jednostkowych i integracyjnych), mający na celu zapewnienie bezawaryjności tej wersji aplikacji.	
10.	Wprowadzenie błędów wersji piaskownicy	Identyfikator: 1201
	Wersja sklepu będąca piaskownicą ma zawierać umyślnie wprowadzone błędy pozwalające na naukę testowania aplikacji. Co jest głównym celem naszej aplikacji.	

Tabela 3: wymagania niefunkcjonalne

Wymagania Niefunkcjonalne		
1.	Wykorzystanie Python <i>Django</i> REST <i>Framework</i>	Identyfikator: 2101
	Wykorzystanie <i>Django</i> REST <i>Framework</i> do budowy interfejsu API umożliwiającego komunikację między klientem a serwerem.	
2.	Wykorzystanie bazy danych SQL-Light	Identyfikator: 2102
	Użycie bazy danych <i>SQLite</i> do przechowywania informacji o produktach, użytkownikach oraz zamówieniach.	
3.	Dostarczenie dokumentacji kodu	Identyfikator: 2103
	Użycie Python <i>DocString</i> do tworzenia dokumentacji kodu w celu ułatwienia zrozumienia i utrzymania aplikacji.	
4.	Dostarczenie dokumentacji testów	Identyfikator: 2301
	Dostarczenie pełnej dokumentacji wykonywanych testów aplikacji referencyjnej to ważny element, spełniający fundamentalny cel projektu.	
5.	Zadbanie o UX/UI	Identyfikator: 2104
6.	Dostarczenie dokumentacji wprowadzonych błędów wersji Piaskownicy	Identyfikator: 2201
	Dostarczenie materiałów opisujących wprowadzone błędy pozwoli użytkownikom końcowym na samo sprawdzenie postępów nauki, oraz ocenę własnych umiejętności.	

Poniżej zaprezentowane została legenda identyfikatorów:

Legenda identyfikatorów – [1][2][3, 4]

- [1]: 1 – Wymaganie funkcjonalne
 2 – Wymaganie нефункционалне
- [2]: 1 – Dla obu wersji
 2- Dla wersji piaskownicy
 3- Dla wersji referencyjnej
- [3, 4]: Numer

Struktura bazy danych

Struktura bazy danych została zaprojektowana w celu przechowywania informacji niezbędnych do funkcjonowania aplikacji sklepu internetowego. Poszczególne tabele reprezentują kluczowe elementy systemu, takie jak użytkownicy, produkty, kategorie, zamówienia oraz statusy zamówień. Ta struktura bazy danych zapewnia podstawę dla działania aplikacji oraz obsługi procesu zakupowego przez użytkowników.

Struktura Bazy Danych:

1. Tabela Użytkownicy (Users):

- Przechowuje dane o użytkownikach platformy.
- Pozwala na identyfikację użytkowników oraz przechowuje informacje takie jak nazwa użytkownika, hasło, adres e-mail, wiek i numer telefonu.

2. Tabela Zamówienia Użytkownika (UserOrders):

- Zawiera identyfikatory zamówień, które zostały złożone przez poszczególnych użytkowników.
- Umożliwia powiązanie zamówień z konkretnymi użytkownikami.

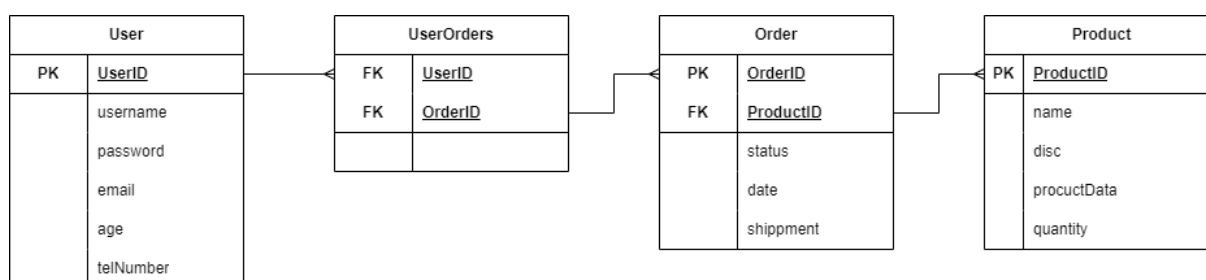
3. Tabela Zamówienie (Order):

- Przechowuje szczegółowe informacje o każdym zamówieniu, takie jak status, data złożenia zamówienia i status dostawy.
- Pozwala na śledzenie zamówień oraz zarządzanie nimi w systemie.

4. Tabela Produkt (Product):

- Zawiera informacje o wszystkich produktach dostępnych w systemie.
- Umożliwia identyfikację produktów oraz przechowuje ich opis i dostępną ilość.

grafika 1: struktura bazy danych



Plan testowania

Z uwagi na specyfikę aplikacji testLearning, która ma na celu naukę testowania oprogramowania, istotnym aspektem projektu jest zapewnienie, że wersja referencyjna aplikacji będzie bezbłędna i kompleksowo przetestowana. Aplikacja ta ma posłużyć jako wzorzec dla zrozumienia procesu testowania aplikacji oraz umożliwić użytkownikom eksperymentowanie z różnymi technikami testowymi.

Wersja referencyjna aplikacji testLearning powinna być kompletna, stabilna i pozbawiona wszelkich błędów, aby stanowiła solidny punkt odniesienia dla osób uczących się testowania oprogramowania. Ponadto, należy zadbać o to, aby wersja referencyjna była bogata w różnorodne testy, które pokrywają różne aspekty funkcjonalne i нефункционалне aplikacji.

Wraz z kompleksowym zestawem testów, wersja referencyjna powinna być także odpowiednio udokumentowana, co umożliwi użytkownikom lepsze zrozumienie testowanych funkcjonalności oraz wyników testów. Dzięki temu, użytkownicy będą mogli bardziej efektywnie korzystać z aplikacji testLearning w celu doskonalenia swoich umiejętności w dziedzinie testowania oprogramowania.

Tu należy wspomnieć że wersja przeznaczona dla użytkownika końcowego nie przewiduje jej wcześniejszego testowania, tak więc ta sekcja dokumentacji odnosić się będzie tylko do wersji referencyjnej programu.

Wykorzystane narzędzia:

W celu zapewnienia wysokiej jakości testów naszego rozwiązania wykorzystane zostały następujące narzędzia:

1. **Pytest:** Jest to popularny framework do testowania aplikacji napisanych w Pythonie. Pytest oferuje intuicyjny interfejs do pisania testów jednostkowych, integracyjnych i funkcjonalnych. Dzięki dużej elastyczności, można go wykorzystać do testowania różnych aspektów aplikacji Django.

2. **Django TestCase:** Wbudowany framework do testowania aplikacji Django. Pozwala on na pisanie testów jednostkowych oraz integracyjnych, które są specyficzne dla Django. Umożliwia testowanie modeli, widoków, formularzy i innych elementów Django.
3. **Selenium:** Jeśli aplikacja zawiera interfejs użytkownika, Selenium może być użyteczne do testowania interakcji użytkownika. Pozwala ono na automatyzację testów przeglądarkowych, co umożliwia symulowanie działań użytkownika i sprawdzanie reakcji aplikacji.
4. **Coverage.py:** Narzędzie do pomiaru pokrycia kodu testami. Jest przydatne do oceny, które części aplikacji zostały przetestowane, a które wymagają dodatkowej uwagi. Dzięki temu można zapewnić, że aplikacja testLearning będzie kompleksowo pokryta testami.

Plan testów:

1. **Cel testowania:**
 - Wszystkie funkcjonalności aplikacji testLearning, zarówno te podstawowe, jak i zaawansowane, będą poddane testom.
2. **Przeprowadzanie testów:**
 - Testy będą przeprowadzane przy użyciu różnych technik testowania, takich jak testy jednostkowe, testy integracyjne, testy funkcjonalne.
 - Testy jednostkowe będą sprawdzały pojedyncze komponenty aplikacji, takie jak modele, widoki, formularze i funkcje.
 - Testy integracyjne będą testować interakcje między różnymi komponentami aplikacji.
 - Testy funkcjonalne będą oceniać, czy aplikacja zachowuje się zgodnie z oczekiwaniami użytkownika.
3. **Momenty wykonywania testów:**
 - Testy będą przeprowadzane na różnych etapach cyklu życia projektu, począwszy od etapu rozwoju, przez testy jednostkowe i integracyjne podczas implementacji, aż po testy funkcjonalne przed wdrożeniem aplikacji.
 - Testy jednostkowe będą wykonywane na bieżąco podczas implementacji poszczególnych komponentów.
 - Testy integracyjne będą przeprowadzane po zakończeniu implementacji poszczególnych modułów.
 - Testy funkcjonalne będą przeprowadzane na zakończenie każdej iteracji lub przed planowanym wdrożeniem.

Strategia testowania:

Testowanie aplikacji testLearning będzie miało na celu zapewnienie, że wersja referencyjna tej aplikacji będzie kompletna, stabilna i pozbawiona błędów. W tym celu testy będą obejmować różnorodne scenariusze użytkowania, aby upewnić się, że wszystkie funkcje aplikacji działają zgodnie z oczekiwaniami. Planowane poziomy testowania obejmą testy jednostkowe, integracyjne, systemowe i akceptacyjne, co zapewni kompleksowe pokrycie aplikacji. Rodzaje testów będą również różnorodne, obejmując testy funkcjonalne, które sprawdzą poprawność funkcji, które ocenią wydajność, bezpieczeństwo i inne aspekty. W tym procesie wykorzystane zostaną różnorodne wspomniane wcześniej narzędzia testowe.

Raportowanie i metryki

Podczas testowania aplikacji testLearning będą zbierane różnorodne dane, które posłużą do oceny jakości systemu. Te dane będą następnie używane do oceny jakości systemu poprzez analizę trendów, identyfikację obszarów wymagających poprawy oraz podejmowanie decyzji dotyczących dalszego rozwoju aplikacji. Poniżej zaprezentowane zostały interesujące nas dane:

1. Wyniki testów: Dokładne wyniki każdego przeprowadzonego testu, wraz z informacją o oczekiwanym i rzeczywistym rezultacie.
2. Pokrycie kodu: Informacje na temat pokrycia kodu przez przeprowadzone testy, które pomogą ocenić, jakie części aplikacji zostały przetestowane, a które nie.
3. Czas wykonania testów: Czas, który był potrzebny na przeprowadzenie wszystkich testów, co może pomóc w identyfikacji ewentualnych problemów z wydajnością aplikacji.
4. Liczba i rodzaj błędów: Dokładny zapis wszystkich napotkanych błędów, wraz z ich rodzajem, priorytetem oraz krokiem reprodukcji.

Przechowywanie i organizacja testów

- Repozytoria kodu: Testy są przechowywane w systemach kontroli wersji, takich jak Git, co umożliwia śledzenie zmian i współpracę zespołową.
- Struktura katalogów: Testy są zorganizowane w katalogach odpowiadających poszczególnym modułom aplikacji (np. order/tests, shop/tests).
- Narzędzia CI/CD: Testy są zintegrowane z narzędziami CI/CD, takimi jak Jenkins czy GitHub Actions, co umożliwia automatyczne uruchamianie testów przy każdej zmianie kodu.
- Konwencje nazewnictwa: Nazwy plików i klas testowych są zgodne z konwencjami przyjętymi w projekcie, co ułatwia nawigację i zrozumienie kodu.

Implementacja testów

1. **Testy formularzy:**
 - **Technologia:** Python, Django.
 - **Lokalizacja:** Pliki testowe w aplikacji Django, np. order/tests.py.
 - **Organizacja:** Testy są zorganizowane w klasy odpowiadające testowanym formularzom.
 - **Konwencje nazewnictwa:** Klasy testowe są nazwane zgodnie ze standardem XxxFormTest, a metody testowe test_XXX.
2. **Testy modeli:**
 - **Technologia:** Python, Django.
 - **Lokalizacja:** Pliki testowe w aplikacji Django, np. register/tests.py.
 - **Organizacja:** Testy są zorganizowane w klasy odpowiadające testowanym modelom.
 - **Konwencje nazewnictwa:** Klasy testowe są nazwane zgodnie ze standardem XxxModelTest, a metody testowe test_XXX.
3. **Testy widoków:**
 - **Technologia:** Python, Django.
 - **Lokalizacja:** Pliki testowe w aplikacji Django, np. shop/tests.py.
 - **Organizacja:** Testy są zorganizowane w klasy odpowiadające testowanym widokom.

- **Konwencje nazewnictwa:** Klasy testowe są nazwane zgodnie ze standardem XxxViewTest, a metody testowe test_xxx.