

Sklep Rowerowy „BikeStrike”

Spis treści

1. Podstawowe informacje o systemie	2
1.1 Przeznaczenie aplikacji.....	2
1.2 Podstawowe funkcjonalności	2
1.3 Użytkownicy docelowi	2
1.4 Obsługa klientów	2
1.5 Wymagania użytkownika	2
2. Diagram przypadków użycia	4
3. Diagram klas – analityczny	5
4. Diagram klas – projektowy	5
5. Tekstowy opis scenariusza „Dodawanie klienta”.....	6
6. Diagram aktywności dla przypadku użycia „Dodawanie zamówienia”	7
7. Diagram stanu dla klasy Zamówienie	7
8. Projekt GUI dla dodawania klienta.....	8
9. Omówienie decyzji projektowych i skutków analizy dynamicznej.	12

1. Podstawowe informacje o systemie

1.1 Przeznaczenie aplikacji

Aplikacja służy do wsparcia operacyjnego sklepu rowerowego "BikeStrike" poprzez rejestrację i monitorowanie kluczowych procesów biznesowych w środowisku stacjonarnym.

1.2 Podstawowe funkcjonalności

System koncentruje się na czterech obszarach:

- Ewidencji personelu sklepowego
- Obsłudze bazy klientów
- Zarządzaniu asortymentem rowerowym
- Przetwarzaniu zamówień handlowych

1.3 Użytkownicy docelowi

Dostęp posiadają wyłącznie zatrudnieni w "BikeStrike" z przypisanymi rolami:

- **Pracownik magazynowy** (dodawanie produktów, aktualizacja stanów magazynowych)
- **Sprzedawca** (obsługa zamówień, rejestracja klientów)
- **Manager** (nadzór nad personelem)

1.4 Obsługa klientów

Typy klientów:

- Klient indywidualny
- Klient firmowy

Wymagane dane:

- Imię i nazwisko (lub nazwa firmy)
- Adres email i telefon
- Data rejestracji w systemie

1.5 Wymagania użytkownika

System przechowuje dane o pracownikach z podziałem na stanowiska:

- Pracownik magazynu
- Sprzedawca
- Manager

System przechowuje informacje o klientach z podziałem na rodzaj:

- Klient indywidualny
- Klient biznesowy

Dane podstawowe dla wszystkich osób w systemie:

- Imię, nazwisko, data urodzenia
- Adres zamieszkania, numer telefonu

Zarządzanie produktami:

- Kategorie: Rowery, Akcesoria
- Atrybuty podstawowe: nazwa, cena, marka, stan magazynowy
- Specyficzne dla rowerów: typ (górski/szosowy/miejski), waga
- Specyficzne dla akcesoriów: rozmiar (np. dla kasków)

Funkcjonalności produktowe:

- Trwałe przechowywanie danych między uruchomieniami systemu
- Możliwość dodawania nowych kategorii produktów

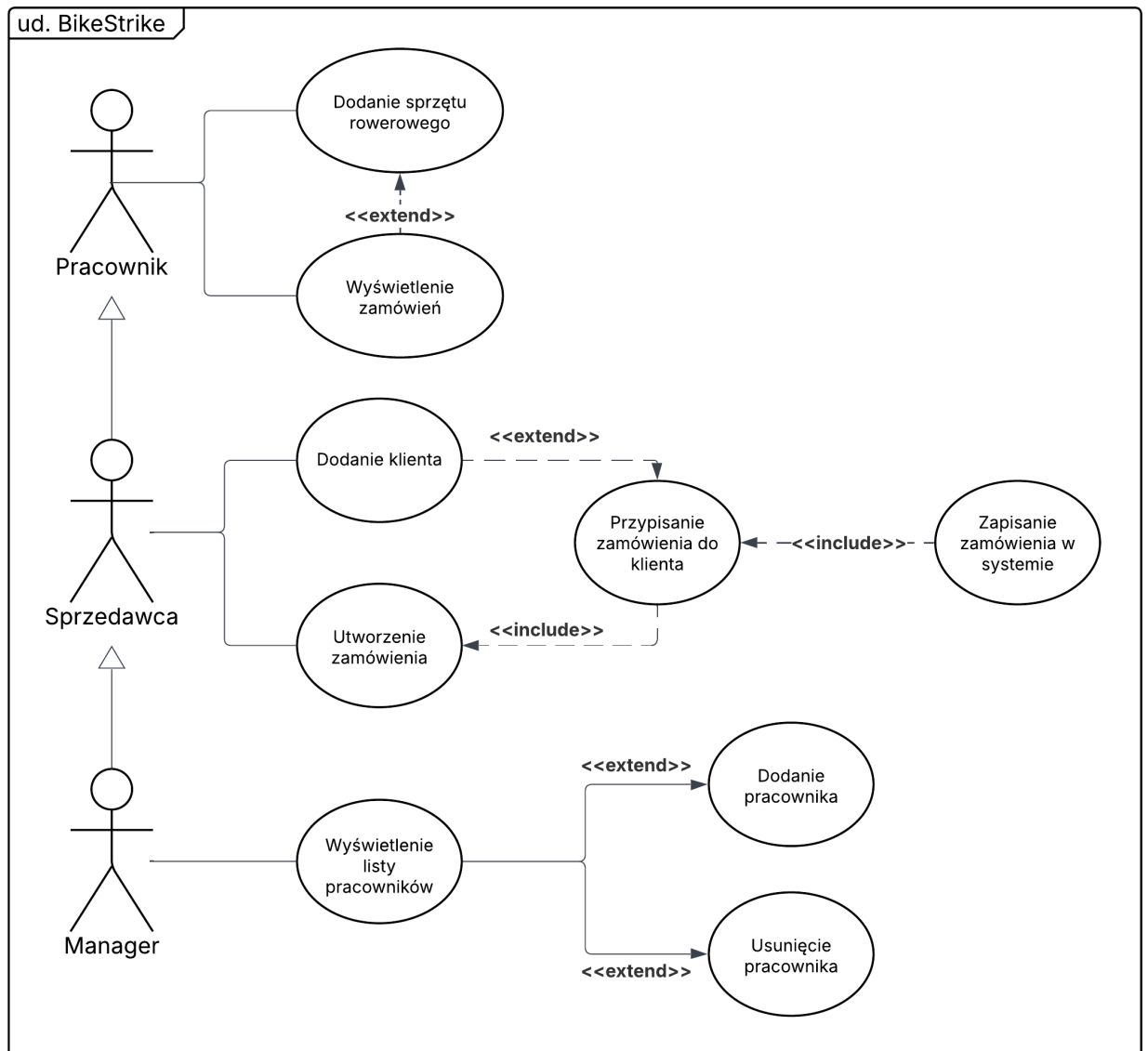
Zarządzanie zamówieniami:

- ID zamówienia, data utworzenia, status (nowe/w realizacji/dostarczone)
- Całkowita wartość zamówienia, data zapisania przez pracownika
- Możliwość przypisania wielu produktów do zamówienia z ilościami
- Przechowywanie pełnych informacji o produktach w zamówieniu

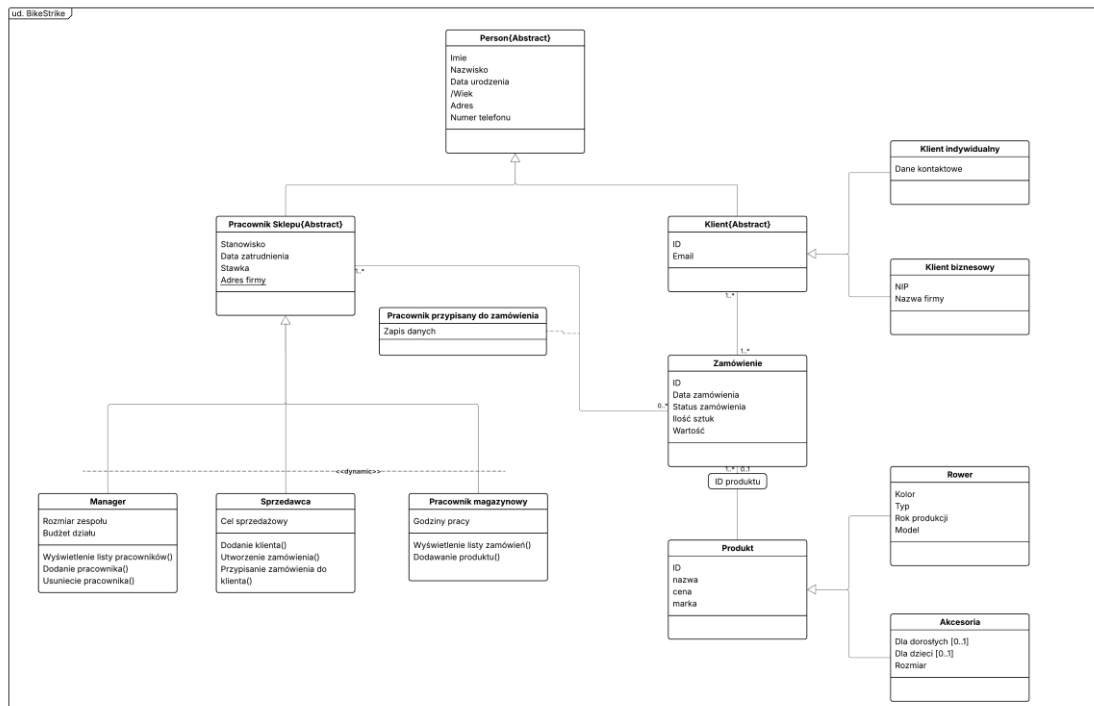
Przypisanie odpowiedzialności:

- Każde zamówienie może być przypisane do wielu pracowników
- Możliwość zmiany przypisania w trakcie realizacji

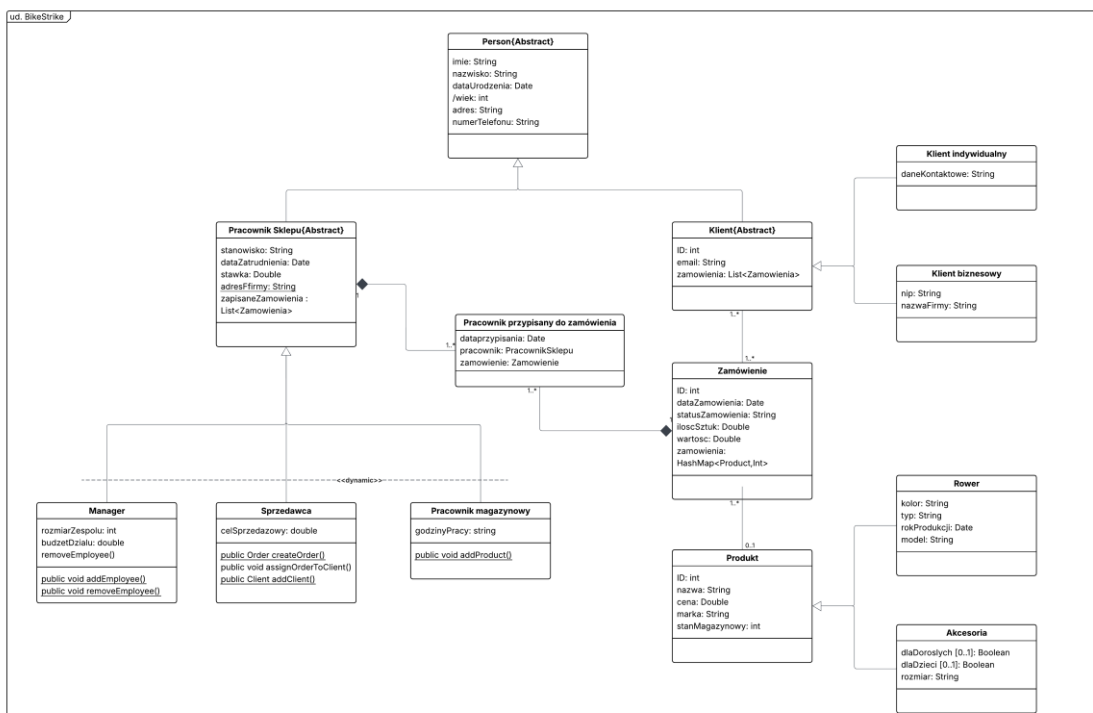
2. Diagram przypadków użycia



3. Diagram klas – analityczny



4. Diagram klas – projektowy



5. Tekstowy opis scenariusza „Dodawanie klienta”.

Aktor główny: Sprzedawca

Warunek wstępny: Sprzedawca jest zalogowany do systemu BikeStrike

Przebieg główny:

1. Sprzedawca wybiera opcję "Dodaj klienta" w systemie
2. System wyświetla formularz dodawania klienta
3. Sprzedawca wypełnia dane formularza:
 - Dla klienta indywidualnego: imię, nazwisko, email, telefon
 - Dla klienta biznesowego: nazwa firmy, NIP, email
4. System sprawdza, czy klient już istnieje w bazie danych
5. System weryfikuje poprawność wprowadzonych danych
6. Jeśli dane są poprawne i klient nie istnieje w systemie:
 - System generuje unikalny identyfikator klienta
 - System zapisuje nowego klienta w bazie danych
7. System wyświetla komunikat potwierdzający dodanie klienta z wygenerowanym ID
8. Proces kończy się sukcesem

Scenariusze alternatywne:

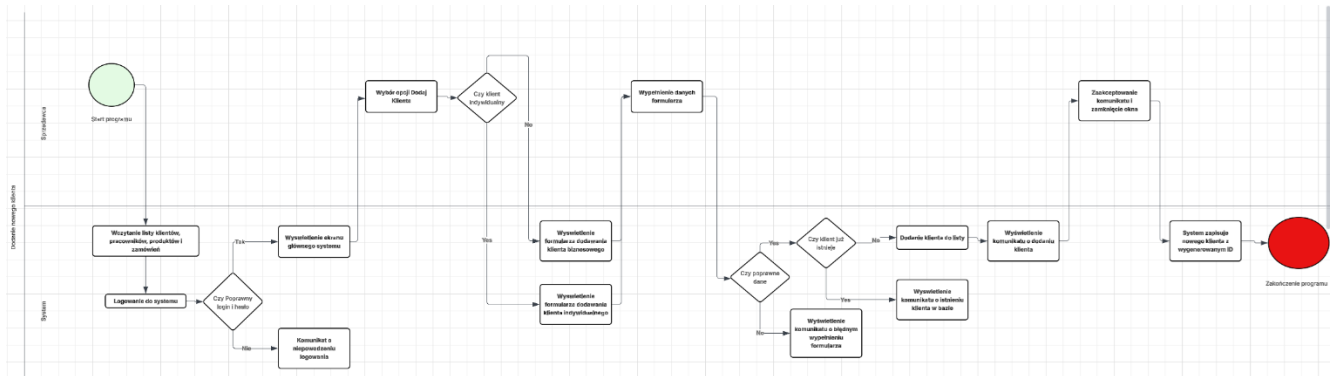
1a. Klient już istnieje w systemie

- System wykrywa, że klient o podanych danych już istnieje
- System wyświetla komunikat: "Klient o podanych danych już istnieje w bazie"
- Sprzedawca akceptuje komunikat
- Proces kończy się bez dodania nowego klienta

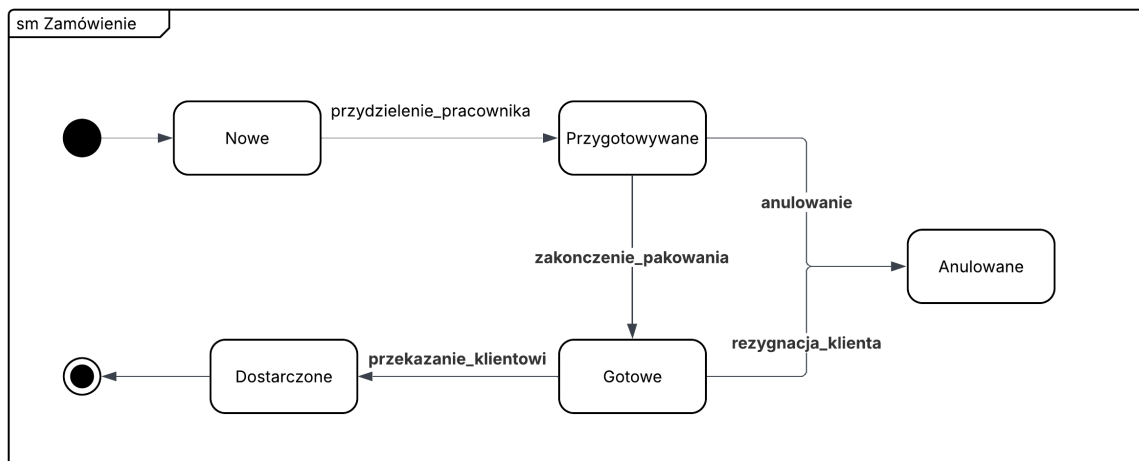
1b. Niepoprawne dane w formularzu

- System wykrywa błędy w danych (np. brak wymaganego pola, niepoprawny format NIP)
- System wyświetla komunikat o błędzie z informacją, które pola wymagają poprawy
- Sprzedawca akceptuje komunikat
- Proces kończy się bez dodania klienta
- Sprzedawca może ponowić próbę z poprawionymi danymi

6. Diagram aktywności dla przypadku użycia „Dodawanie zamówienia”



7. Diagram stanu dla klasy Zamówienie



8. Projekt GUI dla dodawania klienta.

1. Wybór typu klienta



2. Formularz dla klienta indywidualnego

 Dodaj klienta indywidualnego

Imię*:

Nazwisko*:

Data urodzenia (RRRR-MM-DD):

Adres*:

Numer telefonu*:

Email*:

Alternatywne dane kontaktowe:

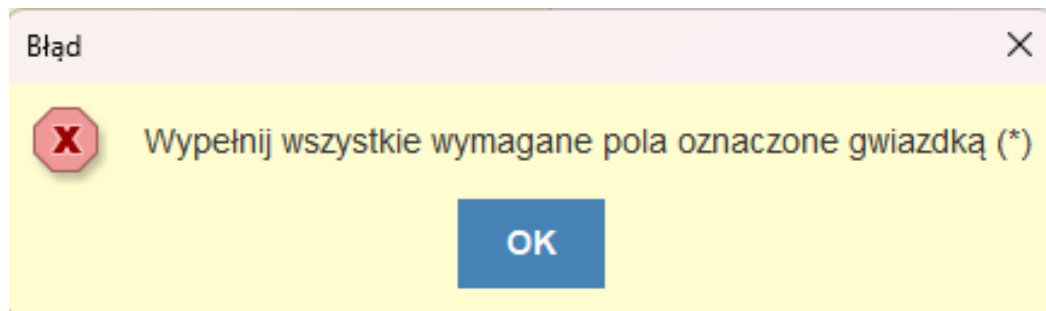
Zapisz

Anuluj

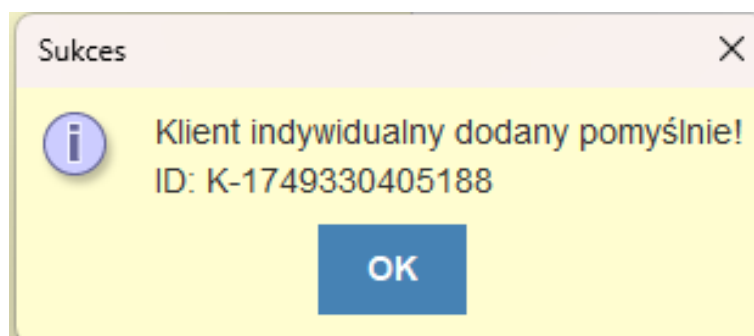
3. Formularz dla klienta biznesowego

Nazwa firmy*:	<input type="text"/>
NIP*:	<input type="text"/>
Imię (osoba kontaktowa)*:	<input type="text"/>
Nazwisko (osoba kontaktowa)*:	<input type="text"/>
Data urodzenia (RRRR-MM-DD):	<input type="text"/>
Adres firmy*:	<input type="text"/>
Numer telefonu*:	<input type="text"/>
Email*:	<input type="text"/>
<div>Zapisz</div> <div>Anuluj</div>	

4. Komunikat o błędzie w formularzu



5. Komunikat o poprawnie dodanym kliencie.



9. Omówienie decyzji projektowych i skutków analizy dynamicznej.

Ekstensje klas:

- List<Pracownik> w klasie Pracownik
- List<Klient> w klasie Klient
- Uzasadnienie: Prostsze zarządzanie instancjami

Zastosowanie HashMap dla produktów w zamówieniu:

- HashMap<Produkt, Integer> w klasie Zamowienie
- Uzasadnienie: Umożliwienie asocjacji kwalifikowanej między zamówieniem i produktem