

Piotr Cebulski

Piotr Jarosik

Anna Stepień

Marcin Wlazły

System do obsługi magazynu

# Spis treści

<b>1. Analiza wymagań</b>	2
1.1. Wstęp	2
1.1.1. Przeznaczenie systemu	2
1.1.2. Architektura systemu	2
1.2. Aktorzy	3
1.3. Wymagania funkcjonalne	4
1.4. Wymagania niefunkcjonalne	6
1.5. Specyfikacja przypadków użycia na poziomie ogólnym	6
<b>2. Przypadki użycia</b>	11
2.1. Moduł zarządzania systemem	11
2.2. Moduł sprzedaży	11
2.3. Moduł zarządzania towarami	11
<b>3. Model analityczny</b>	12
<b>4. Wstępna specyfikacja rozwiązania</b>	13

# 1. Analiza wymagań

## 1.1. Wstęp

Wraz z rozwojem handlu konieczne stało się stworzenie miejsca w którym towary będą składowane przed zakupem przez klienta. W przypadku małych magazynów to pracownicy są w stanie nim efektywnie zarządzać i efektywnie wyszukiwać znajdujące się w nim towary. Jednak co dzieje się w przypadku, gdy liczba towarów przechowywanych w magazynie jest bardzo duża? Konieczne staje się stworzenie systemu który umożliwi pracownikom prostsze i bardziej efektywne zarządzanie towarami znajdującymi się w magazynie.

Głównym zadaniem systemu do zarządzania magazynem jest przechowywanie ilości poszczególnych towarów, odznaczanie dostaw jak i również sprzedaży towarów klientom. Powinien również przechowywać niezbędne informacje o dostawcach, towarach oraz klientach.

### 1.1.1. Przeznaczenie systemu

Głównym zadaniem systemu jest ułatwienie zarządzania magazynem pracownikom magazynu, poprzez umożliwienie zarządzania klientami, dostawcami, towarami a także zamówieniami sprzedaży jak i zakupu. System ten nie jest specjalizowany pod konkretną dziedzinę handlu – ma on umożliwiać przechowywanie i zarządzanie informacjami o towarach dowolnego typu.

### 1.1.2. Architektura systemu

W obecnie tworzonych systemach spotyka się dwie podstawowe struktury: architekturę klient-serwer oraz architekturę trójwarstwową. W architekturze klient-serwer, która była szczególnie popularna w latach dziewięćdziesiątych ubiegłego wieku, wyróżnia się dwie warstwy:

- \* aplikację użytkownika (*klient*);
- \* system zarządzania bazą danych (*serwer*).

Struktura ta sprawdza się dla prostych systemów, których zadaniem jest zapisywanie, odczytywanie oraz aktualizacja danych. Problem pojawia się jednak w przypadku, gdy dane muszą być przetwarzane w nietrywialny spo-

sób, przy uwzględnieniu dziedziny problemu (ang. *domain logic*) modelowanego zagadnienia. Realizacja obliczeń w warstwie klienta, której głównym zadaniem jest prezentacja informacji użytkownikowi, może znacząco wpływać na jego komfort pracy oraz powodować problemy związane z duplikacją kodu źródłowego aplikacji. Natomiast umieszczenie logiki aplikacji po stronie serwera bardzo często narzuca przygotowanie programu w środowisku specyficznym dla danego systemu zarządzania bazami danych.

Problemy te zmusiły projektantów aplikacji do wydzielenia jeszcze jednego poziomu, który jest odpowiedzialny za logikę operacji na danych. W architekturze trójwarstwowej uwzględnione są następujące warstwy:

- \* warstwa prezentacji;
- \* warstwa aplikacji;
- \* warstwa źródła danych.

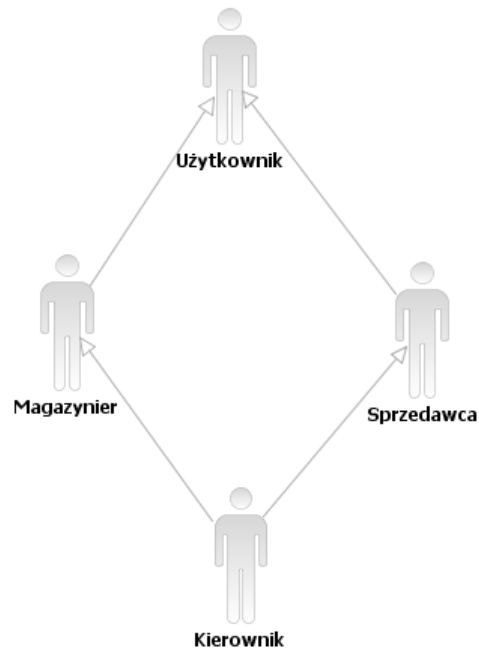


Rysunek 1.1: Schemat architektury trójwarstwowej

W przypadku realizowanego systemu do obsługi magazynu zastosowana będzie architektura trójwarstwowa.

## 1.2. Aktorzy

Aktorzy w zaprojektowanym systemie zostali podzieleni na dwie grupy: aktorów osobowych oraz aktorów nieosobowych. Do aktorów osobowych można zaliczyć wszystkich użytkowników projektowanego systemu, aktorami nieosobowymi są natomiast wszystkie systemy zewnętrzne współpracujące z projektowanym systemem. Poniższy diagram przedstawia hierarchię aktorów osobowych w systemie:



Rysunek 1.2: Diagram przedstawiający aktorów w systemie

### 1.3. Wymagania funkcjonalne

Niniejsze rozdział zawiera wymagania funkcjonalne jakie powinien spełniać projektowany system. Wymagania te zostały podzielone na większe grupy wymagań takie jak: zarządzanie danymi klientów czy zarządzanie danymi dostawców. Grupy wymagań zawierają bardziej szczegółowe wymagania, takie jak edycja czy usuwanie danych. W tabeli umieszczone zostały również informacje o priorytecie wymagania oraz ryzyku i aktorach.

<b>Lp.</b>	<b>Nazwa</b>	<b>Priorytet</b>	<b>Ryzyko</b>	<b>Nazwa aktora</b>
1	Zarządzanie użytkownikami	wysoki	niskie	kierownik
1.1	Dodawanie użytkowników	wysoki	niskie	kierownik
1.2	Edycja danych użytkowników	wysoki	niskie	kierownik
1.3	Usuwanie danych użytkowników	wysoki	niskie	kierownik
2	Zarządzanie danymi towarów	wysoki	niskie	magazynier
2.1	Dodawanie towaru	wysoki	niskie	magazynier
2.2	Edycja danych towarów	wysoki	niskie	magazynier
2.2.1	Edycja opisu towarów	wysoki	niskie	magazynier
2.2.2	Zmiana ilości towarów	wysoki	niskie	użytkownik
2.3	Usuwanie danych towarów	wysoki	niskie	magazynier
3	Zarządzanie danymi klientów	wysoki	niskie	sprzedawca
3.1	Dodawanie klientów	wysoki	niski	sprzedawca
3.2	Edycja danych klientów	wysoki	niskie	sprzedawca
3.3	Usuwanie danych klientów	wysoki	niskie	sprzedawca
4	Zarządzanie danymi dostawców	wysoki	niskie	magazynier
4.1	Dodawanie dostawców	wysoki	niskie	magazynier
4.2	Edycja danych dostawców	wysoki	niskie	magazynier
4.3	Usuwanie danych dostawców	wysoki	niskie	magazynier
5	Zarządzanie zamówieniami sprzedaży	wysoki	niskie	sprzedawca
5.1	Tworzenie zamówień sprzedaży	wysoki	niskie	sprzedawca
5.2	Edycja zamówień sprzedaży	wysoki	niskie	sprzedawca
5.3	Usuwanie zamówień sprzedaży	wysoki	niskie	sprzedawca
5.4	Realizacja zamówień sprzedaży	wysoki	niskie	sprzedawca
5.5	Anulowanie zamówień sprzedaży	wysoki	niskie	sprzedawca
6	Zarządzanie zamówieniami zakupu	wysoki	niskie	magazynier
6.1	Tworzenie zamówień zakupu	wysoki	niskie	magazynier
6.2	Edycja zamówień zakupu	wysoki	niskie	magazynier
6.3	Usuwanie zamówień zakupu	wysoki	niskie	magazynier
6.4	Realizacja zamówień zakupu	wysoki	niskie	magazynier
6.5	Anulowanie zamówień zakupu	wysoki	niskie	magazynier
7	Zarządzanie raportami	średni	niskie	kierownik
7.1	Generowanie raportów sprzedaży	średni	niskie	kierownik
7.2	Generowanie raportów zakupów	średni	niskie	kierownik

## 1.4. Wymagania niefunkcjonalne

Rozdział ten zawiera wymagania niefunkcjonalne dotyczące projektowanego systemu. Wymagania zostały podzielone na kilka grup, przy każdym z wymagań został określony priorytet oraz ryzyko.

Lp.	Nazwa	Priorytet	Ryzyko
1	Bezpieczeństwo	wysoki	niskie
1.1	Mechanizm uwierzytelniania użytkowników za pomocą loginu i hasła	wysoki	niskie
1.2	Zróżnicowanie uprawnień do korzystania z systemu	wysoki	niskie
1.3	Zapisywanie historii przeprowadzanych transakcji	wysoki	niskie
1.4	Szyfrowanie haseł w bazie danych funkcją skrótu	wysoki	niskie
2	Dostępność	wysoki	niskie
2.1	Dostępność systemu co najmniej 98% w skali dnia	wysoki	niskie
2.2	Dostępność przez przeglądarkę	wysoki	niskie
3	Elastyczność	wysoki	niskie
3.1	Architektura systemu umożliwia łatwą rozbudowę i konserwację	wysoki	niskie
3.2	Możliwość integracji z innymi systemami używanymi w firmie	wysoki	niskie
4	Niezawodność	wysoki	niskie
4.1	Maksymalny czas powstania po awarii 2h	wysoki	niskie
4.2	Odtwarzanie uszkodzonych danych na podstawie kopii zapasowych	wysoki	niskie
4.3	Wykonywanie kopii zapasowych danych	wysoki	niskie
5	Użyteczność	wysoki	niskie
5.1	Intuicyjny sposób obsługi systemu	wysoki	niskie
5.2	Przejrzysty system pomocy	wysoki	niskie
6	Wydażność	wysoki	niskie
6.1	Przechowywanie informacji o transakcjach przez 3 lata	wysoki	niskie
6.2	Umożliwienie równoczesnej pracy przez 200 użytkowników	wysoki	niskie

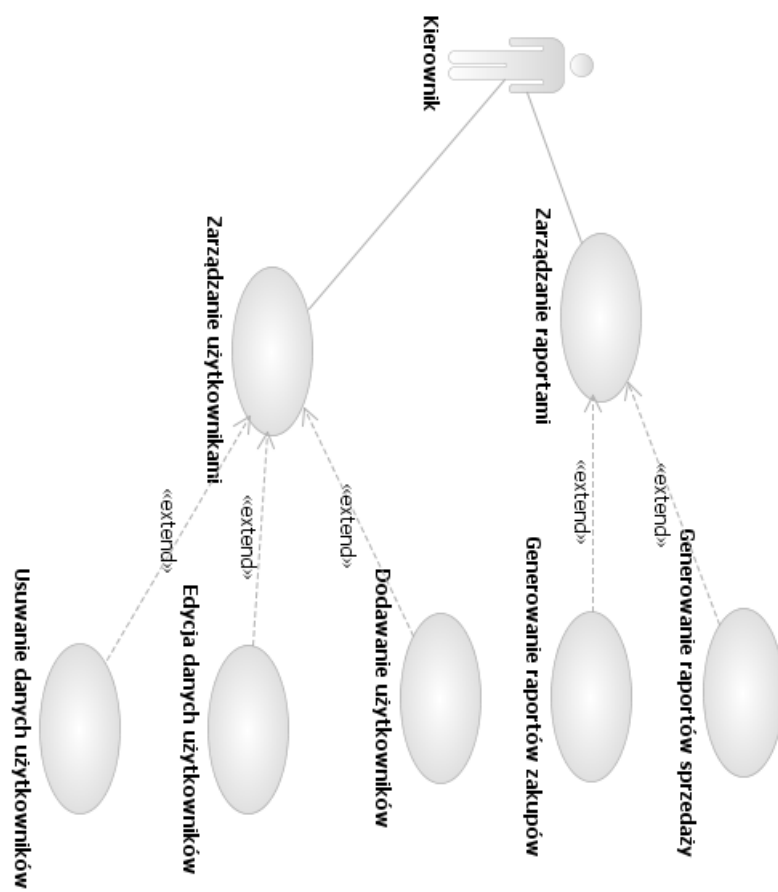
## 1.5. Specyfikacja przypadków użycia na poziomie ogólnym

Dwa podstawowe zadania kierownika to zarządzanie innymi użytkownikami systemu oraz zarządzanie raportami. Posiada on największe uprawnienia w systemie. Może dodawać i usuwać użytkowników, a także nadawać im uprawnienia. Kierownik zajmuje się generowaniem raportów sprzedaży i zakupu. Diagram przedstawiający przypadki użycia kierownika znajduje się na rysunku 1.3.

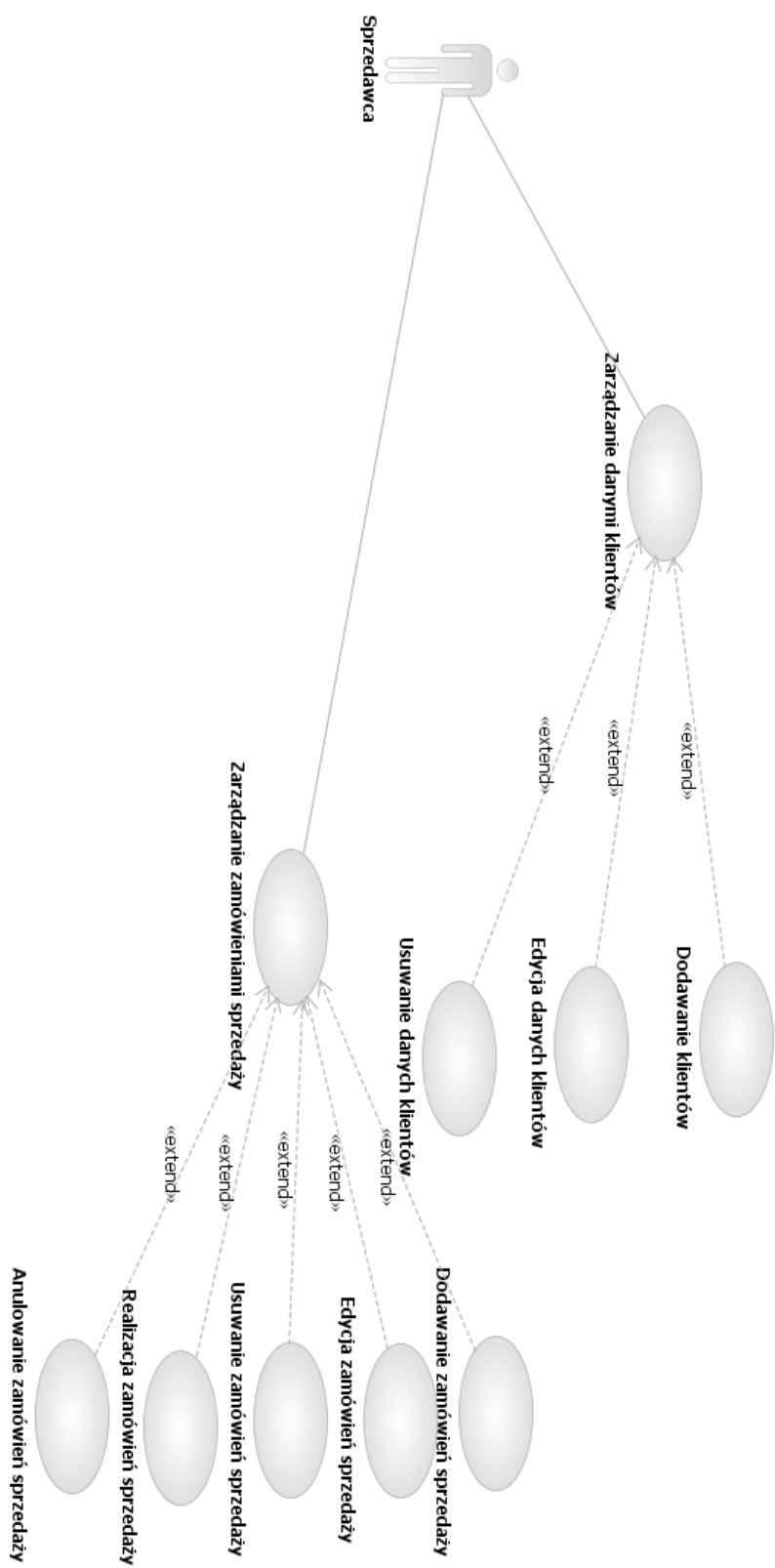
Sprzedawca pośredniczy w kontaktach z klientem i zarządza procesem sprzedaży towarów. Przyjmuje on zamówienia i dodaje je do bazy, aktualizując jednocześnie dane klientów. Sprawuje kontrolę nad realizacją zamówień sprzedaży, a w razie potrzeby może anulować zamówienie. Do zadań sprzedawcy należy prowadzenie ewidencji klientów, aktualizowanie informacji o nich oraz usuwanie starych. Przypadki użycia sprzedawcy zostały zaprezentowane na diagramie 1.4.

Magazynier jest osobą odpowiedzialną za odwzorowanie aktualnego stanu magazynu w systemie. Wprowadza dane towarów do systemu, a także aktualizuje ich ilość. Ponadto zarządza on zamówieniami zakupu oraz danymi dostawców. Zadaniem magazyniera jest prowadzenie ewidencji dostawców, dodawanie nowych oraz usuwanie tych, z którymi zakończono współpracę. Magazynier tworzy zamówienia zakupu zgodnie z zapotrzebowaniem oraz kontroluje realizację zamówień. W razie potrzeby może anulować zamówienie zakupu. Diagram przedstawiający opisane przypadki użycia został zaprezentowany na rysunku 1.5.

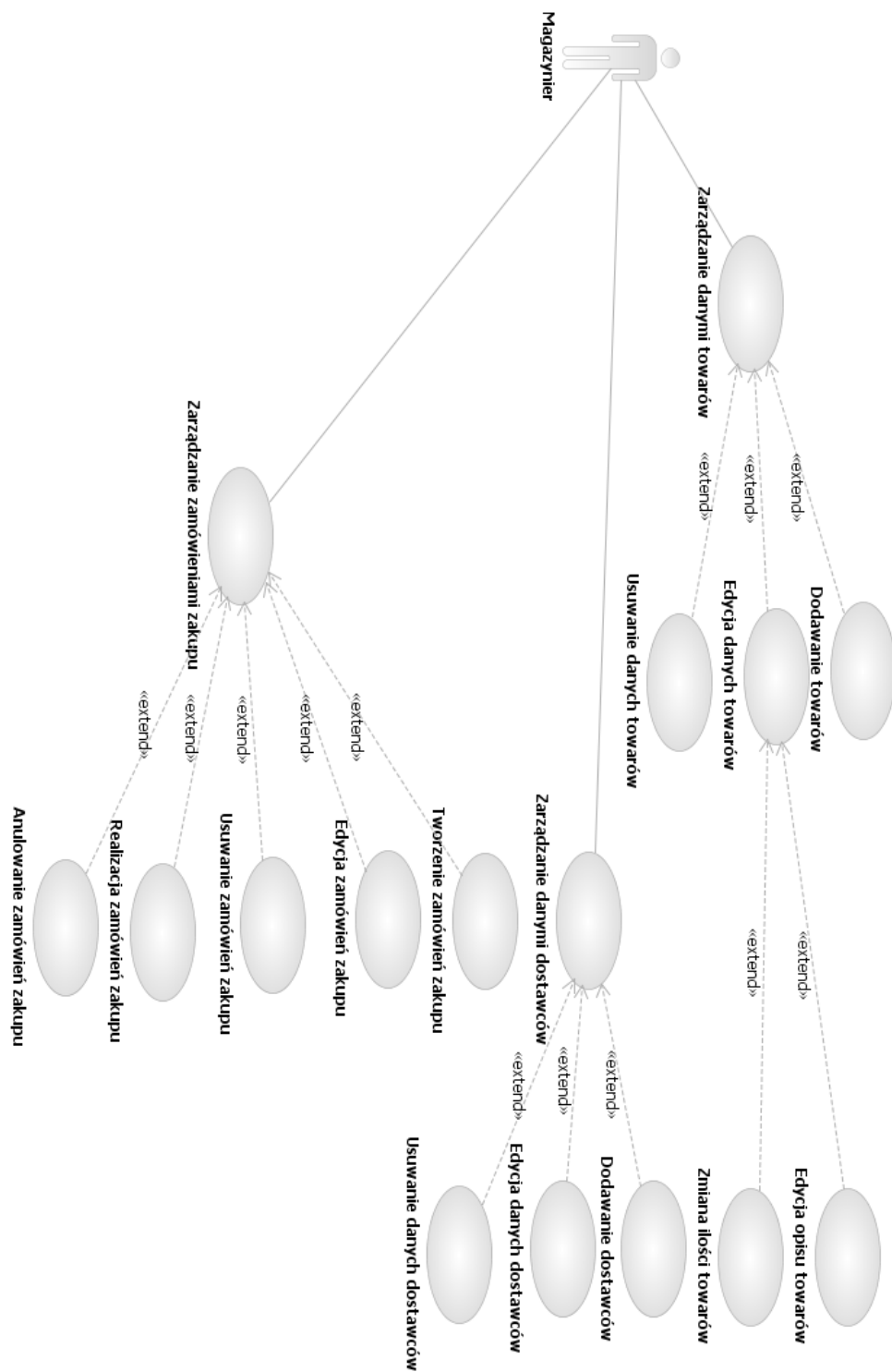




Rysunek 1.3: Diagram przypadków użycia kierownika



Rysunek 1.4: Diagram przypadków użycia sprzedawcy



Rysunek 1.5: Diagram przypadków użycia magazyniera

## 2. Przypadki użycia

### 2.1. Moduł zarządzania systemem

<u>PU1</u>	<u>Generowanie raportów sprzedaży</u>
Priorytet:	średni
Aktor główny:	Kierownik
Warunki początkowe:	<ul style="list-style-type: none"><li>* Kierownik został uwierzytelniony.</li><li>* W systemie przechowywana jest informacja o co najmniej jednej sprzedaży/zakupie.</li></ul>
Warunki końcowe:	Kierownik posiada dokument z raportem z wybranej działalności hurtowni.
Przebieg zdarzeń: <ul style="list-style-type: none"><li>1. Kierownik wybiera opcję tworzenia raportów.</li><li>2. System wyświetla listę możliwych dostępnych typów raportów.</li><li>3.</li></ul>	
Przebiegi alternatywne: <ul style="list-style-type: none"><li>2.a Nie istnieją dane odpowiednie dla danego typu raportu.<ul style="list-style-type: none"><li>1. etc.</li><li>2. etc.</li></ul></li></ul>	
Wymagania funkcjonalne 7. Zarządzanie raportami	

### 2.2. Moduł sprzedaży

Terefere

### 2.3. Moduł zarządzania towarami

Terefere

### 3. Model analityczny

tereferere

## 4. Wstępna specyfikacja rozwiązania

terefere