

MAS-Project

Sprzedaż materacy

Artem Honcharenko(s22527)

# Contents

Dziedzina problemowa.....	3
Cel.....	3
Zakres odpowiedzialności systemu .....	3
Użytkownicy systemu.....	3
Wymagania niefunkcjonalne.....	3
Opis przyszłej ewolucji systemu .....	3
Wymagania użytkownika: .....	3
Diagram przypadku użycia .....	5
Diagram klas – analityczny .....	6
Diagram klas – projektowy.....	6
Scenariusz przypadku użycia .....	7
Diagram aktywności dla przypadku użycia.....	7
Diagram stanu dla klasy .....	8
Diagram sekwencji dla przypadku użycia.....	9
Projekt GUI .....	9
Omówienie decyzji projektowych i skutków analizy dynamicznej.....	10

## Dziedzina problemowa

z tego systemu mogą korzystać firmy zajmujące się sprzedażą dowolnych towarów, ale ten system jest przeznaczony dla firm sprzedających materace.

## Cel

ten system pomoże ułatwić pracę związaną ze sprzedażą, logistyką itp.

## Zakres odpowiedzialności systemu

system ten jest odpowiedzialny za sprzedaż materaców, tworzenie zgłoszeń, wysyłanie ich do magazynu, tworzenie zamówień, przygotowywanie dokumentów i wysyłanie towarów do klientów.

## Użytkownicy systemu

klient, właściciel, dział sprzedaży, pracownicy magazynu, dział logistyki, księgowość

## Wymagania niefunkcjonalne

generalnie system musi pracować prawidłowo, ale mimo wszystko mogą być problemy związane z połączeniem do baz danych dla pracy z Bitrix 24.

## Opis przyszłej ewolucji systemu

być może w przyszłości system automatycznie przesyła dokumenty między działami i automatycznie weryfikuje płatność.

## Wymagania użytkownika:

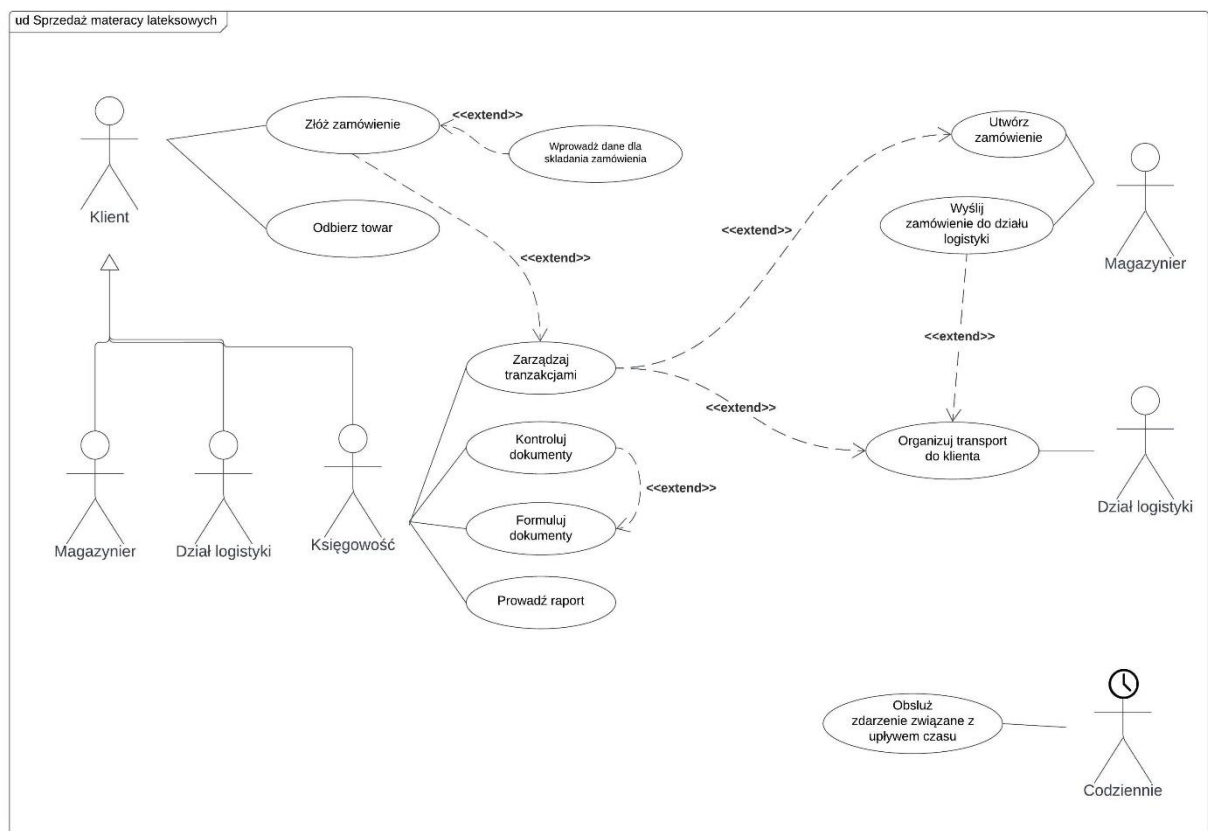
1. W systemie przechowujemy dane pracowników: dane osobowe, data zatrudnienia, adres, stawka dla wszystkich pracowników oraz doświadczenie zawodowe od 2 lat. Są 6 pracowników firmy: logistyk, magazynier, sales manager, buchalter, kierowca oraz ochroniarz. Co miesiąc jest obliczana wypłata, a dla ochroniarza dodatkowo jest uwzględniony dodatek za dodatkowy czas ochrony. Jest możliwość że pracownik może pełnić rolę kierowcy i ochroniarza. Każdy pracownik jest przydzielony do swojego działu.
2. Sklep posiada materacy różnego typu, opisywane przez: rodzaj, cecha, wysokość, sztywność, przeznaczenie, VAT dla wszystkich materacy, cena oraz brutto. Po wybraniu materaca jest wyświetlony cały opis. Materacy są podzielone na: Memory, Siedmiostrefowy oraz Jednostrefowy.
3. Po wejściu na stronę są przechowywane dane klienta: dane osobowe, email oraz opcjonalnie numer telefonu. Klient składa zamówienie jakie przechowuje: data zainicjalizowania, data zrealizowania. Zgłoszenie jest składane za pomocą numeru materaca jaki wybrał klient. Po składaniu rejestruje się zgłoszenie.
4. Po składaniu, zgłoszenie jest zapisane w historię gdzie przechowujemy datę formowania zgłoszenia i datę zakończenia oraz przechodzi do działu sprzedaży gdzie przechowujemy: nazwa produktu, ilość, zleceniodawca, adres, numer telefonu, cena sprzedaży. Dział sprzedaży formuje zgłoszenie

5. Księgowość sprawdza czy istnieje płatność. Jeżeli wszystko w porządku to wysyła fakturę do magazynu. Magazyn tworzy zamówienie i przygotowuje dokumenty. Dział logistyki otrzymuje wszystkie dokumenty i wysyła towar klientowi na jakim jest go adres i data przyjazdu.
6. Dla dostawy w systemie jest wybierany jeden samochód i jeden kierowca i jednym ochroniarzem. Dla każdego transportu przechowujemy takie dane: planowana data rozpoczęcia, faktyczna data rozpoczęcia, planowana data zakończenia, faktyczna data zakończenia, wartość transportu oraz go limit jaki jest dla wszystkich transportów. Gdy wartość transportu przekracza limit to przydzielamy ochroniarza.

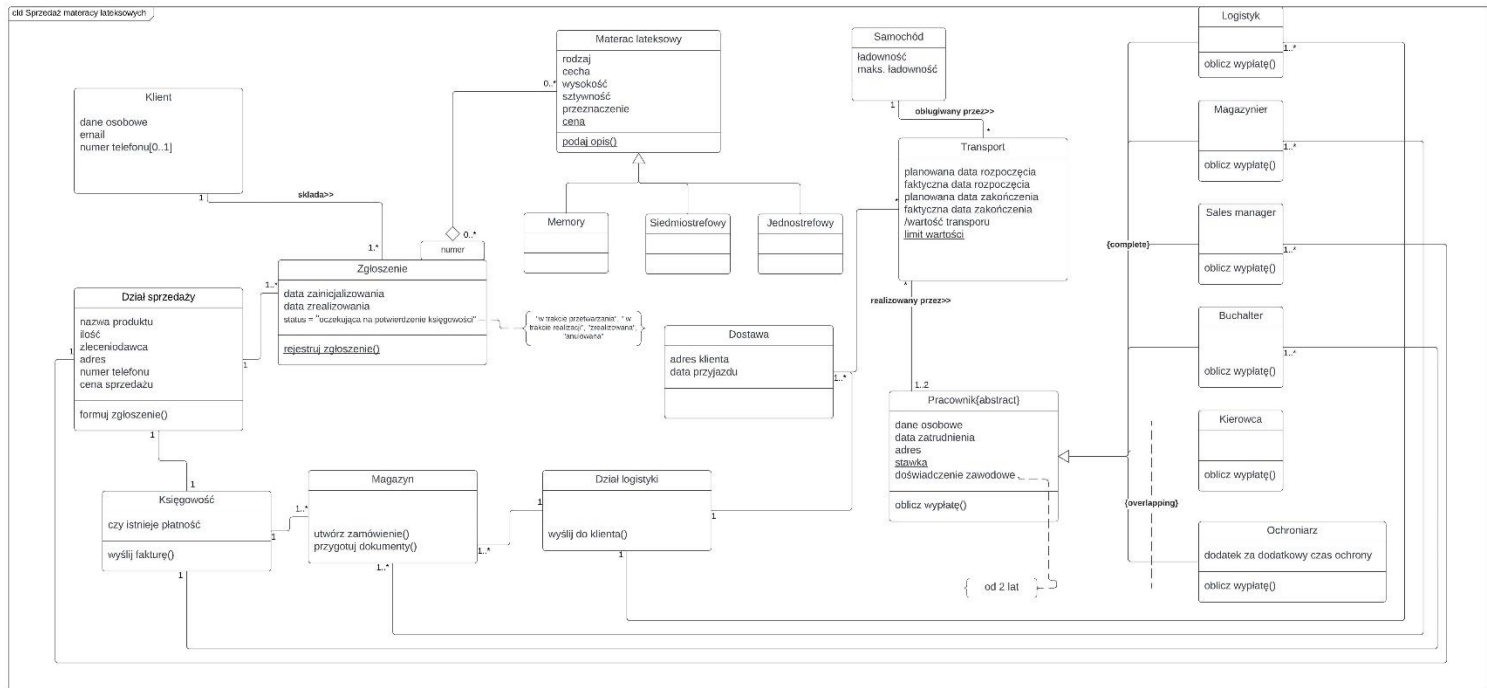
System powinien wspomagać swoich użytkowników w realizacji zadań, takich jak:

- Kontrolowanie, formułowanie dokumentów oraz prowadzenie raportu (księgowość);
- Zarządzanie transakcjami;
- Utworzenie zamówienia oraz wysyłanie go do działu logistyki (magazyn);
- Organizacja transportu do klienta (dział logistyki);
- Złożenie zamówienia, wprowadzenie kryteria dla wyszukiwania pasującego towaru oraz odebranie (klient);
- Obsługiwanie zdarzeń związane z upływem czasu (codziennie).

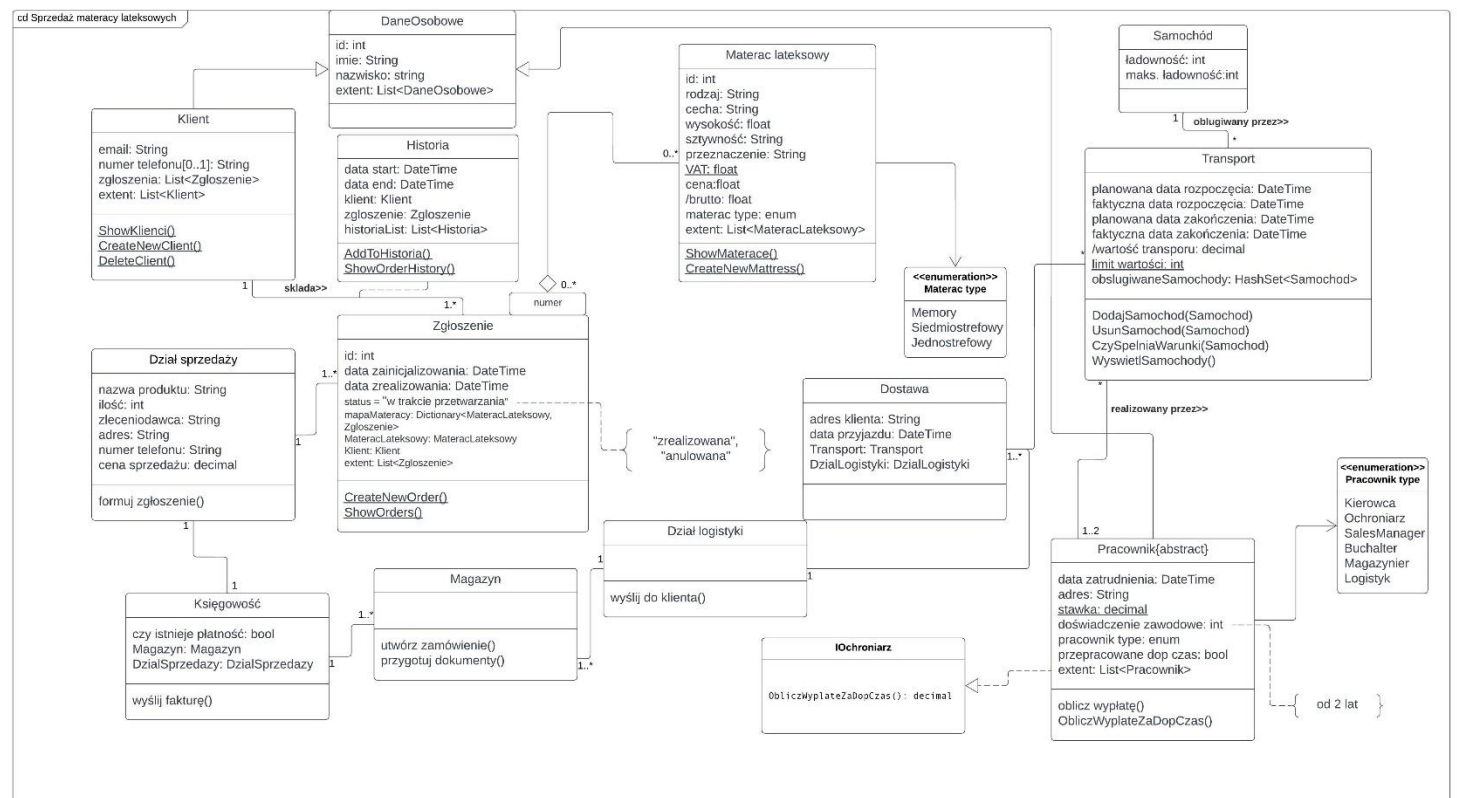
## Diagram przypadku użycia



## Diagram klas – analityczny



## Diagram klas – projektowy



## Scenariusz przypadku użycia

### Sprzedaż materacy lateksowych

**Warunek początkowy:** w systemie jest zarejestrowany co najmniej jeden materac

**Główny przepływ zdarzeń:**

1. Przypadek użycia wywołuje aktor Klient.
2. Klient sprawdza listę materacy
3. Klient składa zamówienie.
4. System prosi wprowadzić dane dla składania zamówienia. Klient wprowadza.
5. Klient wybiera ten materac jaki go interesuje.
6. Po składaniu system wyświetla dane zamówienia

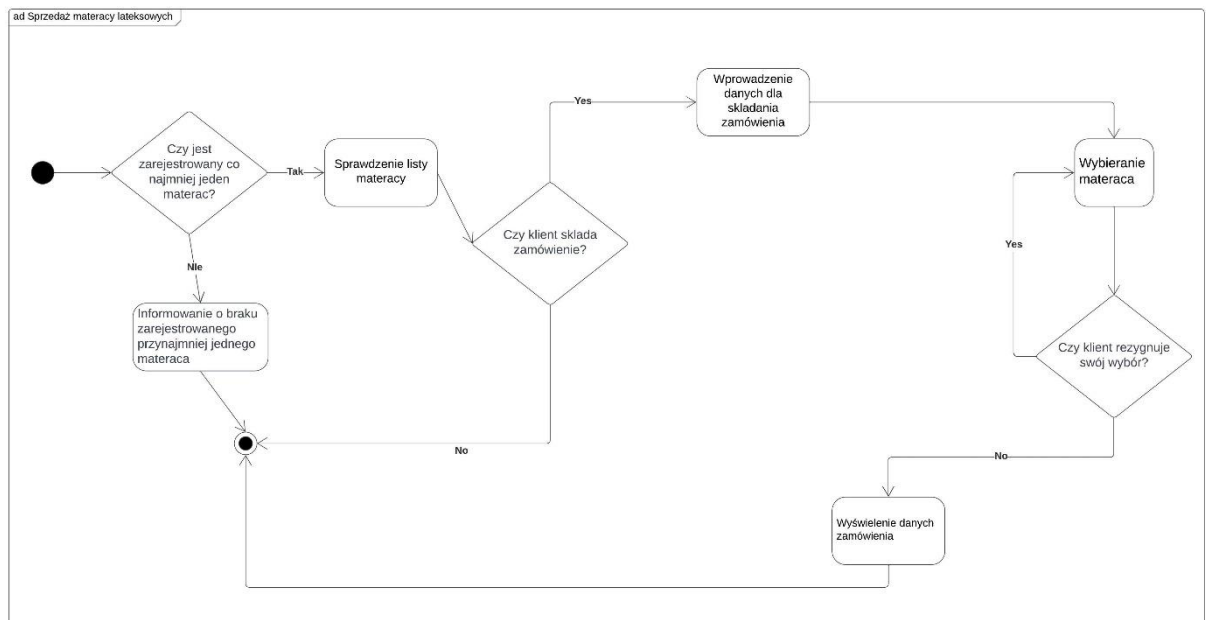
**Alternatywne przepływy zdarzeń:**

3a. Klient nie składa zamówienie.

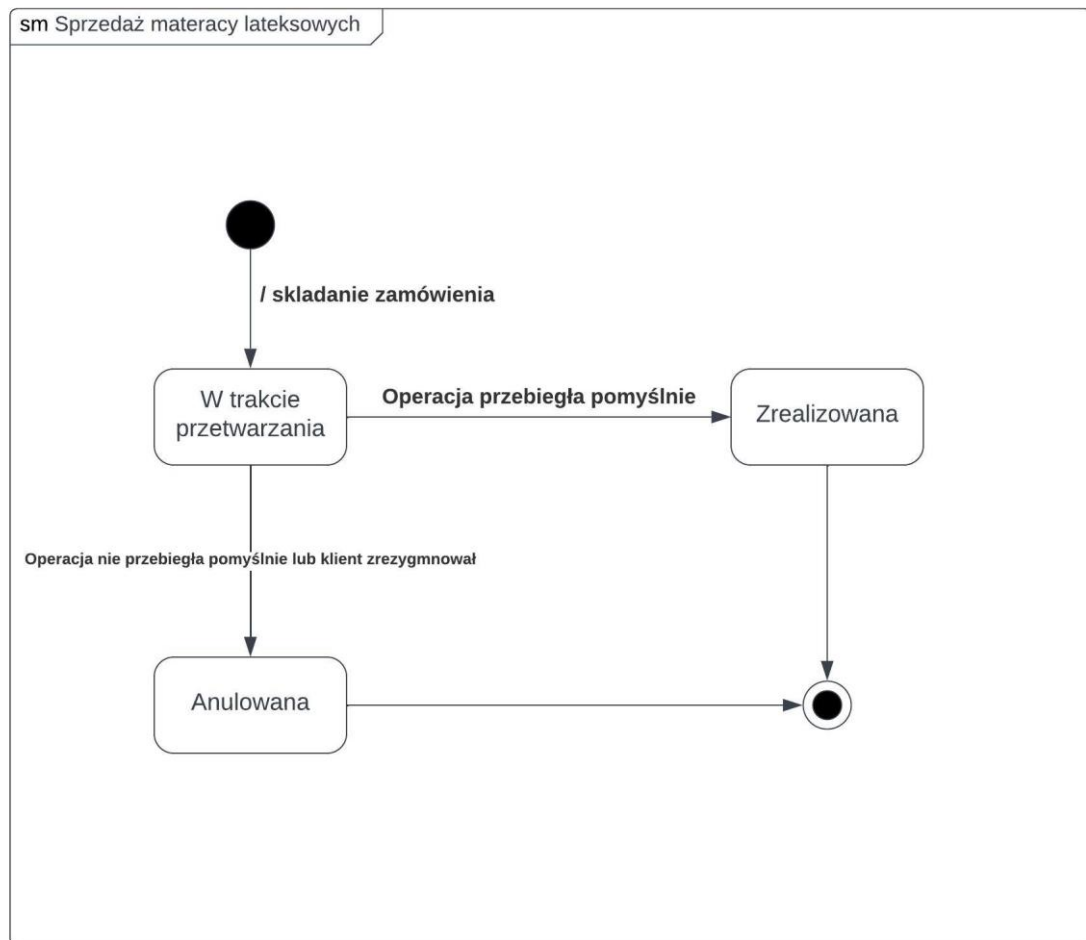
4a. Klient rezygnuje swój wybór. Klient wybiera inny materac.

**Warunek końcowy:** brak

## Diagram aktywności dla przypadku użycia

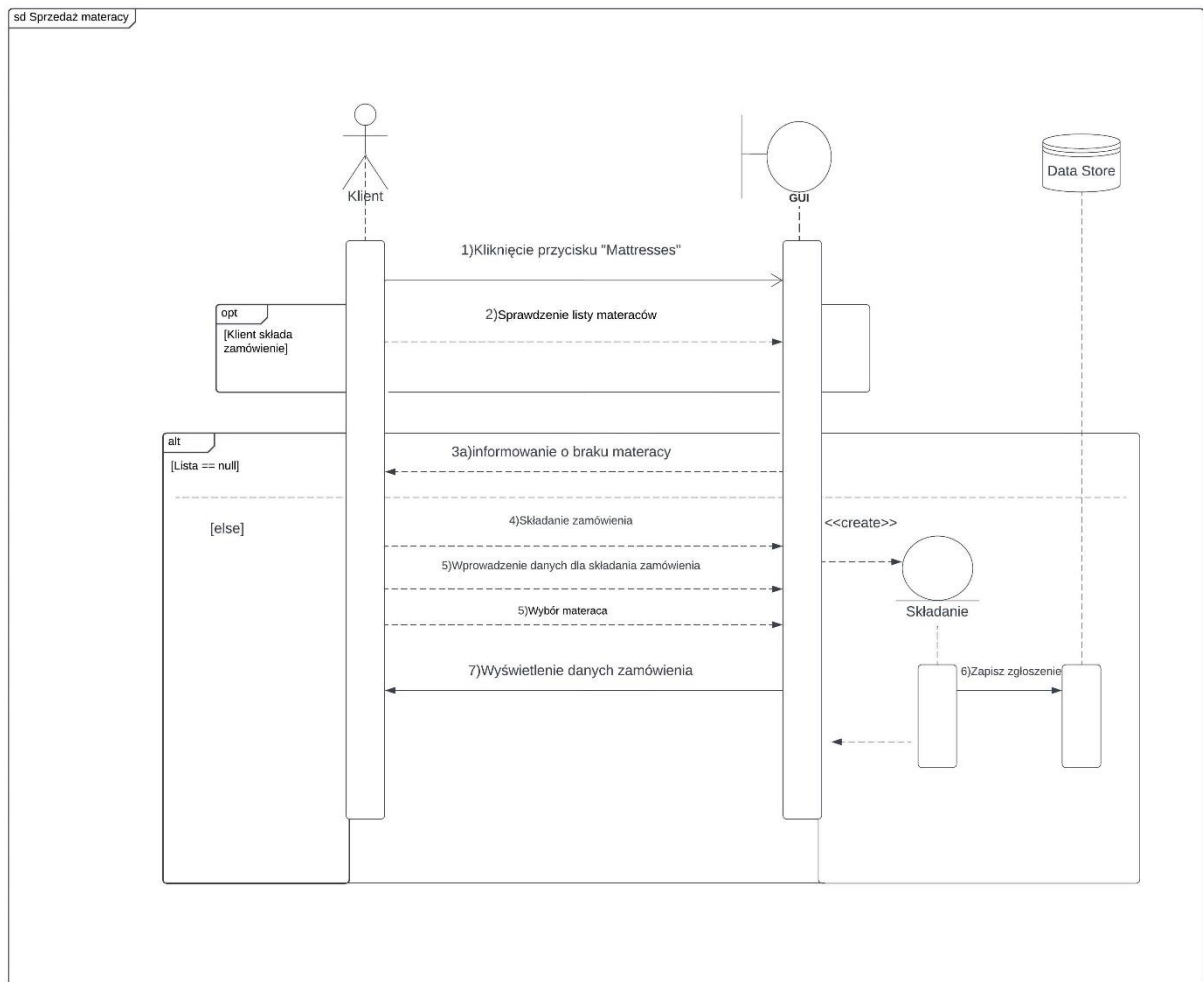


## Diagram stanu dla klasy





## Diagram sekwencji dla przypadku użycia



## Projekt GUI

MAS\_2 Home Privacy Create client Create order

### Create Zgłoszenie

ID

Klient

Имя 1

MateracLateksowy

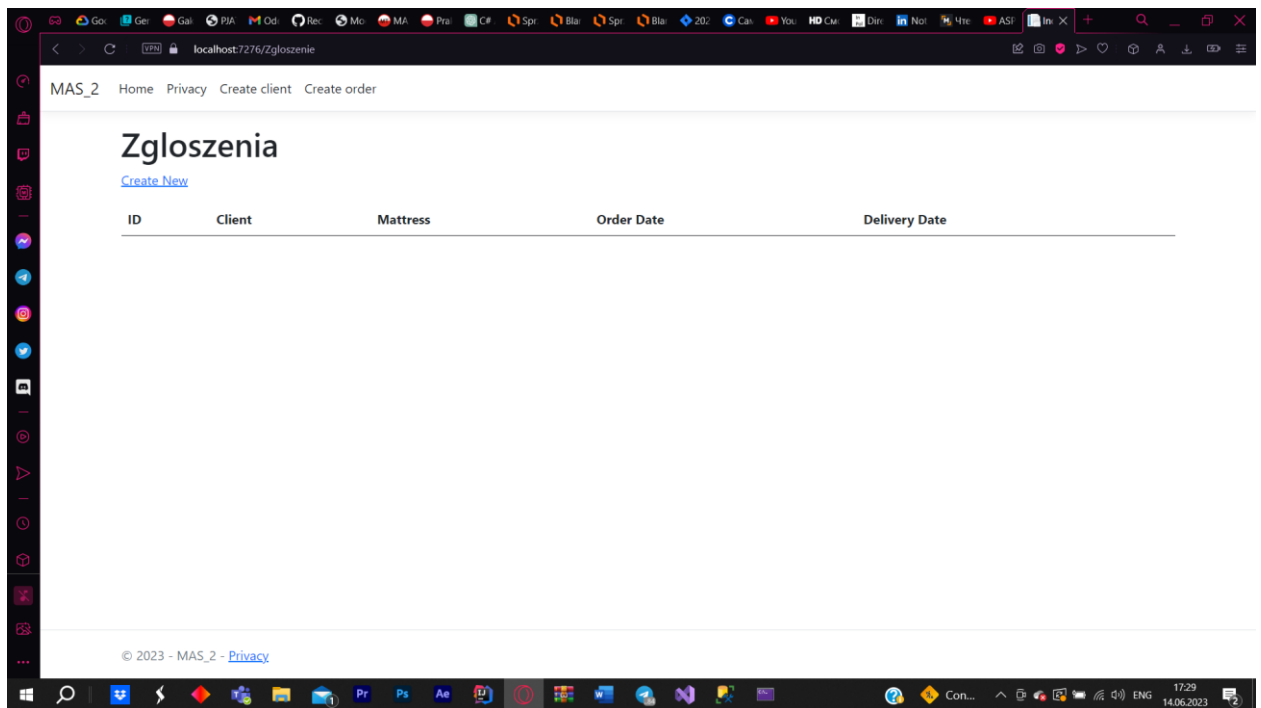
Родзай 1

DataZainicjalizowania

DataZrealizowania

[Back to List](#)

© 2023 - MAS\_2 - Privacy



## Omówienie decyzji projektowych i skutków analizy dynamicznej

### 1. Wybrane decyzje projektowe

#### 1.1 ASP.NET MVC jako wzorzec projektowy

W moim projekcie zdecydowałem się wykorzystać wzorzec projektowy MVC (Model-Widok-Kontroler). MVC pomaga w separacji logiki biznesowej, warstwy prezentacji i zarządzania danymi. Dzięki temu aplikacja staje się bardziej modułowa, elastyczna i łatwa do zarządzania.

#### 1.2 Wybór języka programowania C#

Do implementacji logiki biznesowej w moim projekcie wybrałem język programowania C#. C# jest językiem silnie typowanym, obiektowym i zapewnia wiele zaawansowanych funkcji, które ułatwiają rozwój aplikacji. Ponadto, C# jest integralną częścią platformy .NET, co umożliwia korzystanie z bogatego zestawu bibliotek i narzędzi do tworzenia aplikacji.

#### 1.3 Baza danych SQL Server

Jako system zarządzania bazą danych wybrałem SQL Server. SQL Server jest potężnym DBMS, który zapewnia nie tylko niezawodne przechowywanie danych, ale także zaawansowane funkcje związane z transakcjami, skalowalnością i bezpieczeństwem. Wybór SQL Server pozwala efektywnie zarządzać danymi aplikacji.

## **1.4 Migracje bazy danych**

Aby utrzymać spójność struktury bazy danych z modelem danych w projekcie, zdecydowałem się wykorzystać migracje bazy danych. Migracje pozwalają w łatwy sposób wprowadzać zmiany w strukturze bazy danych i utrzymywać jej wersjonowanie. Dzięki temu można bezpiecznie i łatwo aktualizować bazę danych wraz z rozwojem aplikacji.

## **2. Analiza dynamiczna plików cshtml**

W projekcie korzystam z plików cshtml do tworzenia interfejsu użytkownika. Pliki cshtml pozwalają na dynamiczne generowanie treści HTML, wykorzystanie składni Razor oraz interakcję użytkownika z aplikacją. Analiza dynamiczna plików cshtml obejmuje:

### **2.1 Generowanie dynamicznego HTML**

Dzięki składni Razor, pliki cshtml umożliwiają generowanie dynamicznego HTML w oparciu o dane pobrane z bazy danych lub dostarczone przez kontrolery. Możemy wstawiać zmienne, pętle, warunki i inne konstrukcje programistyczne, co pozwala na elastyczne tworzenie interfejsu użytkownika.

### **2.2 Przekazywanie danych do kontrolerów**

Pliki cshtml pozwalają na przekazywanie danych z formularzy lub innych interakcji użytkownika do kontrolerów. Dzięki temu można odbierać dane wprowadzone przez użytkownika, przetwarzać je i podejmować odpowiednie akcje w aplikacji.

### **2.3 Obsługa zdarzeń użytkownika**

Za pomocą skryptów JavaScript lub wbudowanych funkcji Razor, pliki cshtml umożliwiają obsługę zdarzeń użytkownika, takich jak kliknięcie przycisku, wysłanie formularza, nawigacja między stronami itp. Można reagować na te zdarzenia i wywoływać odpowiednie funkcje w kontrolerach lub modyfikować stan interfejsu użytkownika.

### **2.4 Wyświetlanie danych z bazy danych**

Pliki cshtml pozwalają na wyświetlanie danych pobranych z bazy danych. Możemy używać pętli i warunków, aby dynamicznie generować treść HTML na podstawie danych z bazy danych. Dzięki temu możemy prezentować użytkownikowi informacje związane z zamówieniami, klientami itp.

### 3. Użycie w projekcie dodatkowych elementów

W projekcie wykorzystano także następujące elementy:

- **Asocjacja z atrybutem:** Wykorzystanie atrybutów w klasach do określania asocjacji między różnymi klasami lub encjami w modelu danych.
- **Kompozycja:** Tworzenie hierarchii obiektów, gdzie obiekty podrzędne są związane z obiektem nadrzędnym i zależą od niego.
- **Asocjacja kwalifikowana:** Określanie dodatkowych atrybutów lub kluczy dla asocjacji między klasami, które nadają im dodatkowe znaczenie lub ograniczenia.
- **Atrybut opcjonalny:** Ustalanie, że pewne atrybuty mogą mieć wartość null lub nie być obecne w obiekcie.
- **Ekstensja:** Określanie, jak długo trwa istnienie obiektów w systemie.
- **Atrybut złożony:** Wykorzystywanie atrybutów składających się z kilku elementów, które reprezentują bardziej złożone informacje.
- **Atrybut powtarzalny:** Określanie, że pewne atrybuty mogą mieć wiele wartości, nie tylko pojedynczą.
- **Atrybut klasowy:** Przypisywanie atrybutów do klas, które mają wpływ na wszystkie instancje tych klas.
- **Atrybut pochodny:** Tworzenie nowych atrybutów na podstawie istniejących atrybutów lub innych czynników.
- **Metoda klasowa:** Tworzenie metod, które są związane z klasami, a nie z konkretnymi instancjami tych klas.
- **Przesłonięcie, przeciążenie:** Zmiana zachowania lub dodawanie nowych funkcji do dziedziczonej klasy przez podklasę.
- **Wielodziedziczenie:** Tworzenie hierarchii klas, w której podklasa dziedziczy cechy po więcej niż jednej klasie nadrzędnej.

### 4. Podsumowanie

Decyzje takie jak wybór wzorca MVC, języka C# oraz bazy danych SQL Server miały kluczowy wpływ na architekturę i funkcjonalność aplikacji. Analiza dynamiczna plików cshtml pozwala generować dynamiczny interfejs użytkownika, przekazywać dane do kontrolerów, obsługiwać zdarzenia użytkownika i wyświetlać dane z bazy danych. Dodatkowo, w projekcie wykorzystano elementy takie jak asocjacja z atrybutem, kompozycja, asocjacja kwalifikowana, atrybuty opcjonalne, ekstensja, atrybuty złożone, atrybuty powtarzalne, atrybuty klasowe, atrybuty pochodne, metody klasowe, przesłonięcie, przeciążenie oraz wielodziedziczenie. Dzięki tym decyzjom projektowym i analizie dynamicznej, aplikacja jest elastyczna, skalowalna i dostarcza bogate funkcje dla użytkowników.