

# Food ordering

SEMESTRÁLNY PROJEKT Z ASWS 2023

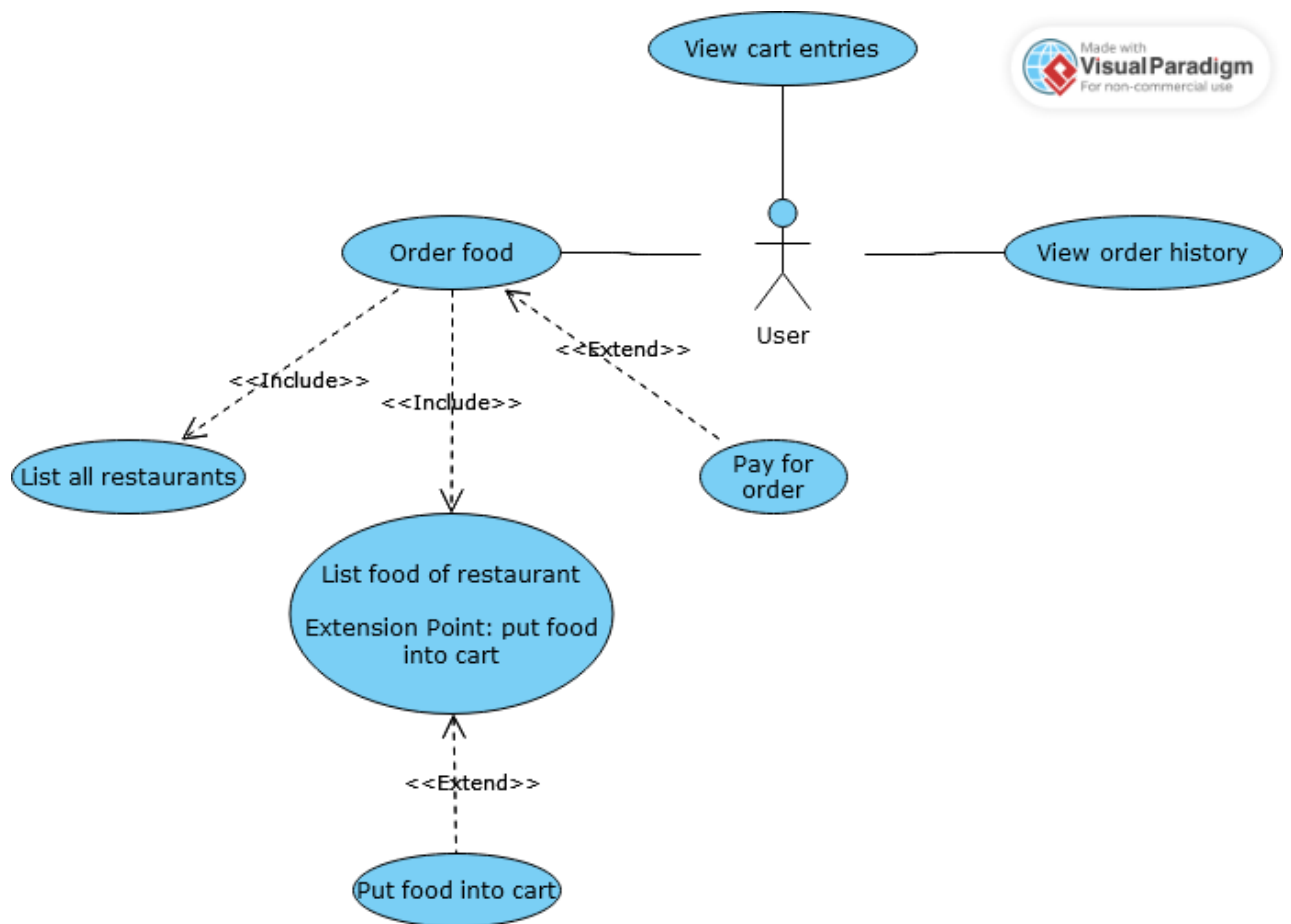
BC. URBANEK RASTISLAV, FMFI UK, 8.6.2023

## 1. Úvod

Aplikácia slúži ako trenažér na .NET a architektonické štýly. Jedná sa o jednoduchú CLI aplikáciu pripojenú na Postgresql databázu. Aplikácia umožňuje používateľovi vyberať si jedlá z obchodov, tvoriť objednávky a prezerať históriu objednávok

## 2. Prípady použitia

### 2.1. Diagram



## 2.2. Scenáre

Aktér: Používateľ

Popis: Prezretie košíka

Tok scenára:

1. Používateľ si v menu zvolí príslušný príkaz
2. Systém vypíše obsah košíka
  - a. Systém oznámi, že obsah košíka je prázdny

Aktér: Používateľ

Popis: Prezretie histórie objednávok

Tok scenára:

1. Používateľ si v menu zvolí príslušný príkaz
2. Systém vyzve používateľa aby vybral, ktorú objednávku si chce prezrieť
3. Používateľ zvolí objednávku
  - a. Systém oznámi, že bola zvolená neexistujúca objednávka
4. Systém vypíše obsah objednávky

Aktér: Používateľ

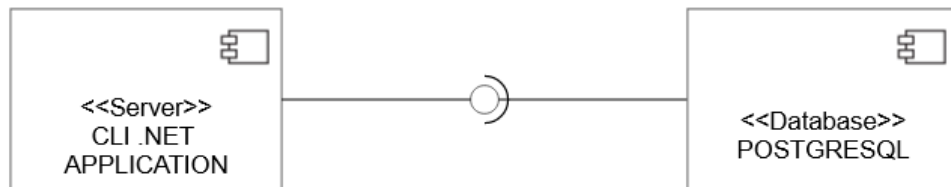
Popis: Objednanie jedla

Tok scenára:

1. Používateľ si v menu zvolí príslušný príkaz
2. Systém vyzve používateľa aby vybral z ktorej reštaurácie jedlo chce
3. Používateľ zadá reštauráciu
  - a. Systém oznámi, že bola zvolená neexistujúca reštaurácia
  - b. Systém oznámi, že bola zvolená reštaurácia ktorá už má zatvorené (????)
4. Systém vyzve používateľa, aby si vybral jedlo z ponuky reštaurácie
5. Používateľ zadá jedlo
  - a. Systém oznámi, že bola zvolená neexistujúca jedlo
6. Systém vyzve používateľa k platbe
7. Používateľ zaplatí
  - a. Používateľ nezaplatí, no môže neskôr

### 3. Architektúra systému

CLI klient komunikujúci cez EF kontext a db drajver s databázou



## 4. Analýza

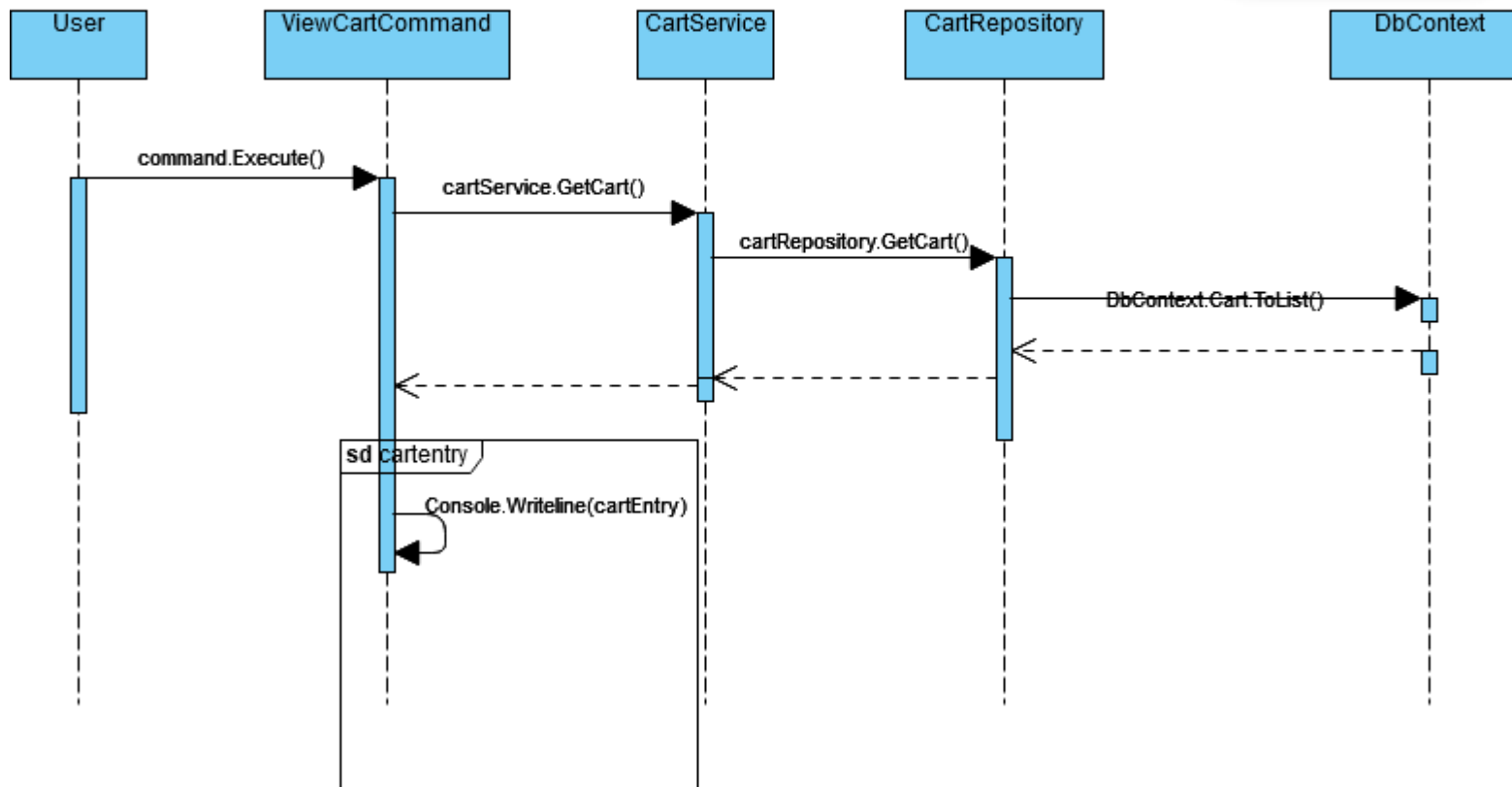
Mali by ste analyzovať doménu, pôvodný informačný systém v nej, ak existuje, ale vy budete v dokumente uvádzať hneď novonavrhnuté efektívne scenáre a štruktúru systému

v nasledovných diagramoch:

### 4.1. Sequence Diagram

(min. 2x), popisujúce scenár prípadov použitia pomocou interakcií medzi objektami, identifikujeme tak potrebné objekty-triedy a ich metódy, nezabudnúť na Loop, Alt, Opt a ich constrainty





4.2. Activity diagram

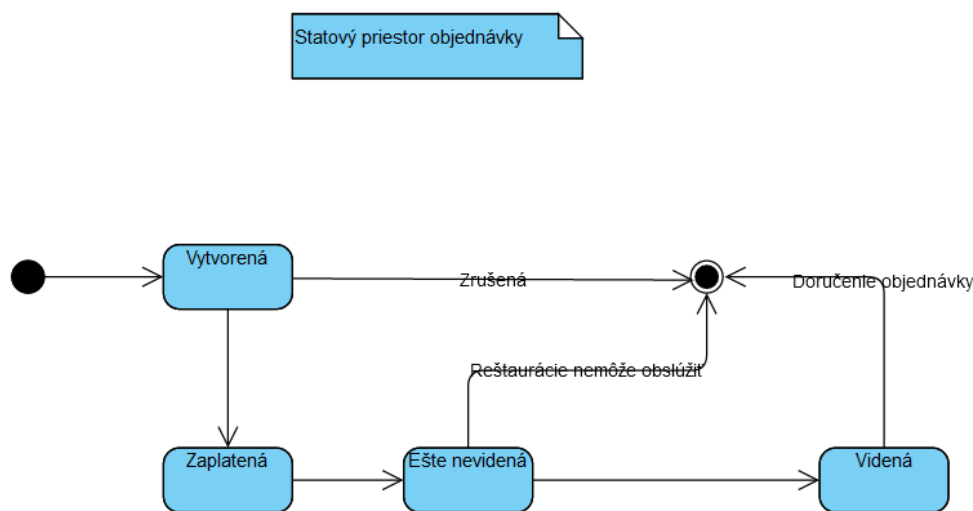


### 4.3. Activity diagram

(min. jeden) pre vybraný scenár alebo algoritmus. Nemiešať elementy decision-merge (pre alternatívne vetvy) a fork-join (pre paralelné vetvy). Použite swimlanes, ale nie paušálne pre celý systém, ale pre jednotlivé objekty/triedy

### 4.4. State diagram

(min. jeden). Uviesť pre ktorú konkrétnu triedu modelujeme stavový priestor a uviesť jej metódy. V samotnom diagrame nezabudnúť udalosti/podmienky na hranách, operácie/metódy triedy najmä v stavoch, menej na hranách



## 5. Návrh a implementácia, identifikácia vzorov

### 5.1. Class Diagram

odvođený od scenárov (SqD a AcD), potrebných objektov a metód. Pri agregácii a dedení nie je potrebné pomenovať väzby, pri klasickej asociácii uviesť názov vzťahu a kardinalitu, prípadne role objektov a pod.; pre atribúty môžete typ, pri metódach môžete zoznam parametrov, ich typ, návratovú hodnotu, nemusíte. Bude to jeho výsledná forma aj s identifikovanými vzormi ktoré budú identifikované v ofarbených poznámkach pri triedach (rola:vzor).

5.2. model (class diagram) implementovaného vzoru,

kde triedy majú dôležité metódy a atribúty +

krátky popis + fragment zdrojáku vzoru z vašej jednoduchkej funkčnej aplikácie aspoň v cmdline,

## COMMAND

```
public interface ICommand
{
    [1 usage] [4 implementations] [Rastislav Urbanek]
    void Execute();

    [1 usage] [4 implementations] [Rastislav Urbanek]
    void Undo();

    [1 usage] [4 implementations] [Rastislav Urbanek]
    string Description { get; }
}
```






```
8 ^ | public class OrderFoodCommand : ICommand
9     {
10         private readonly IStoreService _storeService;
11         private readonly ICartEntryService _cartEntryService;
12         private readonly IStoreFoodItemService _storeFoodItemService;
13         private readonly IOrderService _orderService;
14
15         [1 usage] [Rastislav Urbanek]
16         public OrderFoodCommand(IStoreService storeService, IOrderService orderService,
22
23         [0+1 usages] [Rastislav Urbanek]
24         public void Execute(){...}
46
47         [1 usage] [Rastislav Urbanek]
48         private void OrderFood(){...}
63
64         [1 usage] [Rastislav Urbanek]
65         private Store? SelectStore(){...}
88
89         [1 usage] [Rastislav Urbanek]
90         private List<StoreFoodItem> SelectStoreFoodItems(int storeId){...}
```


## DTO


```
public class Store
{
    [Key]
    public int Id { get; set; }
    public List<StoreFoodItem> StoreFoodItems { get; set; } = new();
    public string Name { get; set; } = string.Empty;
    public TimeSpan OpeningTime { get; set; }
    public TimeSpan ClosingTime { get; set; }
}

public bool IsOpen()
{
    var now:TimeSpan = DateTime.Now.TimeOfDay;
    return now >= OpeningTime && now <= ClosingTime;
}
```

## Context Object

```
public class MyDbContext : DbContext
{
     3 usages
    public DbSet<CartEntry> Cart { get; set; }
     1 usage
    public DbSet<Store> Store { get; set; }
     1 usage
    public DbSet<StoreFoodItem> StoreFoodItem { get; set; }
    public DbSet<FoodItem> FoodItem { get; set; }
     2 usages
    public DbSet<Order> Order { get; set; }
     1 usage
    public DbSet<OrderStoreFoodItem> OrderStoreFoodItem { get; set; }


     Rastislav Urbanek
    > protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder){...}



     Rastislav Urbanek
    > protected override void OnModelCreating(ModelBuilder modelBuilder){...}
```

## Service

```
public class CartEntryService : ICartEntryService
{
    private readonly ICartEntryRepository _cartEntryRepository;

     Rastislav Urbanek
    public CartEntryService(ICartEntryRepository cartEntryRepository)
    {
        _cartEntryRepository = cartEntryRepository;
    }

     0+2 usages  Rastislav Urbanek
    public List<CartEntry> GetCart()
    {
        return _cartEntryRepository.GetAllCartEntries();
    }

     0+1 usages  Rastislav Urbanek
    public void AddToCart(List<StoreFoodItem> selectFoodItemsOfStore)
    {
        _cartEntryRepository.AddToCart(selectFoodItemsOfStore);
    }
}
```

5.3. implementácia vzorov v scenároch prípadov použitia: skopírujete SqD + poznámky ako v 5.1.