

UML del I – Tværfaglig Opgave

Case: Big Mamma pizzeria

I den første version af systemet bør du fokusere på den del af systemet, der omhandler **ordrehåndtering af pizzaer, hvor kunderne kan bestille Pizzaer.**

Du skal arbejde med og gennemføre følgende del opgaver:

1. User Stories	Du skal udforme mindst 10 simple User Stories for 'the Big Mamma pizzeria'. Disse User Stories er i en product backlog
2. Domain Model	Du skal udvikle og tegne en domain model (hvor der mindst er 5 konceptuelle klasser) for 'the Big Mamma pizzeria.' Fremgangsmåde: i. Find de konceptuelle klasser ii. Tegn dem i en UML domain model iii. Tilføj associationer og multipliciteter iv. Tilføj attributer
3. Design Class Diagram	Du skal udarbejde et Design Class Diagram (DCD) for 'the Big Mamma pizzeria' der bygger på din domain model a. Find metoder b. Overvej og indfør retning på de forskellige associationer.
4. Interaction diagram	Du skal udarbejde et interaction diagram (sekvens diagram) for programmet (program.cs), dvs der oprettes 3 pizzaer, 3 kunder og 3 ordrer.

I den første del skal du implementere et IT-system (Console application), hvor alle associationer er 1:1 forbindelser, f.eks.: Én kunde har én ordre med én pizza.

<p>5. Implementer Design Class</p>	<p>A. Opret en Console application PizzaStore</p> <p>B. Implementer alle klasserne fra dit Design Class Diagram.</p> <ul style="list-style-type: none">a. Hver klasse har instance fields, properties, konstruktør samt en ToString-metode <p>C. Opret en metode, '<i>CalculateTotalPrice</i>', der udregner den totale pris for en ordre (selvom der i første version kun er én pizza i en ordre). Metoden skal have en returværdi, der indeholder den totale pris for en ordre dvs. pizzaens pris + moms + leveringsomkostninger (40 kr).</p> <p>D. For at prøve om din applikation virker skal du i 'program.cs'</p> <ul style="list-style-type: none">a. Oprette 3 Pizza objekter, 3 Kunde objekter and 3 Ordre objekter hver med forskellige pizza objekter. <p>E. Udskriv på skærmen information om en ordre</p> <ul style="list-style-type: none">a. For hver Ordre-objekt udskriv navnet på pizzaen, navnet ,på kunden, samt den totale pris.b. Prøv metoden <i>CalculateTotalPrice</i>
------------------------------------	---

Udvidelse af 'The Big Mamma pizzeria'

I denne opgave del skal du redesigne din **Domain Model** (vis nødvendigt), dit **Design Class Diagram** fra opgaven ovenover.

Du skal implementere en ny version af applikationen 'The Big Mamma pizzeria'.

Denne version af systemet skal indeholde:

- Kunde administration
 - Systemet skal kunne indeholde en liste af kunder (slags kunde katalog).
 - Systemet skal kunne oprette en kunde, slette en kunde og opdatere en kunde.
 - Systemet skal kunne søge efter kunder ud fra et kunde navn.

- Pizza menu-kort administration
 - Systemet skal kunne oprette en pizza, slette en pizza og opdatere en pizza.
 - Systemet skal kunne udskrive en liste af pizzaer på skærmen.
 - Systemet skal kunne søge på en pizza ud fra Pizza nummer og udskrive den fundne pizza på skærmen.

6. User Stories	Du skal opdatere dine user stories i din Product Backlog med nye User Stories. Dvs der skal være User Stories med Kunder og Pizza menu-kortet. Desuden skal alle User Stories have et ' Acceptance Criteria '
7. Opdater Domain Model and Design Class Diagram	Du skal opdatere dine diagrammer fra opgaven ovenover, så pizza (evt. toppings), menu-kort, ordre, kunder fremgår af diagrammerne.
8. Udarbejd et Sekvens Diagram	Design og tegn et sequence diagram for metoden der søger efter en bestemt pizza (ud fra et pizzanummer).
9. Implementering	Du skal implementere kundeadministrationen, dette skal gøres med en liste. Du skal implementere Menukort administrationen, dette skal gøres med en Dictionary.

Vink: Gode råd til fremgangsmåde.

Implementer alle 1:* og *:.* associations/aggregations/compositions ved at benytte collections klasser (list, dictionary, ...).

Trin:

- Implementer en klasse KundeKatalog – brug **List**
 - Opret en klasse KundeKatalog, der indeholder Kunder (du burde have en klasse Kunde fra opgaven ovenover), som bliver gemt i en **Liste**,
 - KundeKataloget skal have metoder til CRUD operationer (Create, Read, Update and Delete).
 - Implementer en ToString metode, der skal samle alle kunder i en string.

- Implementer en klasse MenuKort – brug **Dictionary**
 - Opret en klasse MenuKort, der indeholder Pizzaer (du burde have en klasse Pizza fra opgaven ovenover), som bliver gemt i en **Dictionary**, hvor pizzanummeret er 'key' mens pizzaobjektet er 'value',
 - MenuKort skal have metoder til CRUD operationer (Create, Read, Update and Delete).
 - Implementer en metode SearchPizza(criteria), der søger efter en pizza og returnere den fundne pizza.
 - Implementer en ToString metode, der skal samle alle pizzaer i en string.

- I din Applikation (program.cs) skal du prøve dine to nye klasser:
 - Opret et objekt af KundeKatalog.
 - Vis hvordan du tilføjer, sletter og opdaterer kunder i KundeKataloget, dvs. vis hvordan du kalder og benytter metoderne.

 - Opret et objekt af MenuKort.
 - Vis hvordan du tilføjer, sletter og opdaterer pizzaer i MenuKortet,.
 - Vis hvorledes du kan udskrive hele MenuKortet.
 - Vis hvorledes du kan benytte metoden SearchPizza (Du skal **ikke** have input fra tastaturet)

Extra Opgave – Hvis tiden tillader det

Opret en skærm menu.

- Opret en bruger dialog til dit system. Opret en metode MenuValg (Menu Choice), der udskriver brugerens valgmuligheder på skærmen, samt indlæser brugerens menu valg F.eks.:

1. Tilføj en ny pizza
2. Slet en pizza
3. Opdater en pizza
4. Søg efter en pizza
5. Vis alle pizzaer

.....

Metoden kunne se således ud:

```
int MenuChoice( List<string> menuItems) {  
}
```

Metoden skal kunne håndtere fejl fra brugeren.
Metoden skal returnere brugerens valg som et tal.

Benyt denne metode I din applikation (program.cs).

Extra-extra opgaver

- Implementer en forbedret udgave af klassen Ordre og overvej hvorledes en ordre kan indeholde flere pizzaer.
- Implementer en metode der gør det muligt at tilvælge extra toppings til pizzaen.
- Implementer en forbindelse til en kunde.
- Implementer metoder til CRUD for en Ordre.