

Project Report

Mario's Pizza

DAT21b

Project Participants:

Dannie Heegaard Kattrup

Danny Lu

David Borja-Jakobsen

Umaid John Shirzad

19 / 10 - 2021

Document Version: 00.01.00

Document Template version: 00.04.00

Indholdsfortegnelse.

Document Version	2
Vision	3
Business Rules / Domain Rules	4
Requirements	4
Supplementary Specification	5
FURPS	6
Use Cases	7
System Sequence Diagram	12
Operation Contracts	13
Design Class Diagram	14
Sequence Diagram	15
Domain Diagram	17
Glossary	18

Document Version

Date	Version	Author	Description
11/10/2021	00.01.00	Dannie H.K. Danny L. David B.J. Umaid J.S.	Use Cases System Sequence Diagram Domain Model
12/10/2021	00.02.00	Dannie H.K. Danny L. David B.J. Umaid J.S.	FURPS Use Cases
13/10/2021	00.03.00	Danny L.	Oversættelse af begreber
19/10/2021	00.04.00	Dannie H.K. Danny L. David B.J. Umaid J.S.	Added diagrams Added glossary list Removed comments Added Visio

Vision

Projektet omhandler et bedre overblik over salg af pizzaer. Dette skal gøres gennem et program på butikkens bærbar, og gøre det muligt for "cashieren" at behandle ordren af pizza, samtidig med at "chefen" kan se hvad der skal laves og danne sig et overblik af salget.

Dette kan foregå over telefonen, kunde besøg i butikken, og senere hen med henblik på salg online.

Programmet skal indeholde menukortet som skal være tilgængelig hele tiden. Det skal kunne vise aktive ordre og gemme det til senere at kunne lave en statistik over de pizza'er som bliver solgt mest.

Det skal også være muligt at kunne få sig et overblik over programmet ved en hjælpefunktion, som gør det nemt at finde de kommandoer man har brug for.

Samtidig danner det et overblik over hvilke pizza'er som er bestilt, hvor længe det tager og hvad totalprisen er for salget. Efter dette er gjort, gemmer den ordren til senere at danne statistik.

I fremtiden vil ideen være at butikken også går online, men muligvis også en opgradering af setuppet og muligvis wifi i butikken. Det vil gøre det muligt at debugge og opdaterer programmet over nettet frem for fysisk fremmøde. Et bedre setup kunne evt. Være en bedre bærbar eller tablet at køre programmet over og/eller en extra monitor i back office der gør det muligt for Mario (eller andre som er i backoffice) at følge med uden at skulle gå ud til bærbaren og se med på skærmen.

Business Rules / Domain Rules

Rule List

ID	Rule	Changeability	Source
RULE1	Pizza er tilknyttet til en bestilling		
RULE2	Afhentningstidspunkt er tilknyttet til en bestilling		
RULE3	Kunden er tilknyttet til en bestilling		
RULE4	Kunden er tilknyttet til butikken		

Requirements

- REQ001: se en liste med alle *bestillinger*
- REQ002: se en liste med *aktive bestillinger*
- REQ003: hvornår de skal afhentes
- REQ004: gemme alle ordrene
- REQ005: indtast ordrene
- REQ006: hele mit menukort skal på skærmen
- REQ007: fjerne den fra listen, når den er afhentet og betalt
- REQ008: sortere ordrene efter tidspunkt
- REQ009: tager imod *bestillinger*

Supplementary Specification

FURPS

Functional

Id	Requirement	Comment
FR001	Programmet skal kunne køre på en gammel Dell computer	
FR002	Se en liste med bestillingerne	
FR003	Se menukortet på skærmen	
FR004	Sortere bestillinger efter tidspunkt	
FR005	Fjerne bestillinger fra listen	Fjernes når bestillingen er betalt.
FR006	Gemme alle bestillinger	Bestillinger gemmes til omsætning.

Usability

Id	Requirement	Comment
UR001	Fysisk manual	Manual til hvordan programmet bruges
UR002	Digital manual	Evt. en “help” kommando, der udskriver manualen i konsolen
UR003	Kursus til hvordan programmet bruges	En introduktion til programmet

Reliability

Id	Requirement	Comment
RR001	Programmet skal kunne køre hele dagen uden at crashe	
RR002	...	

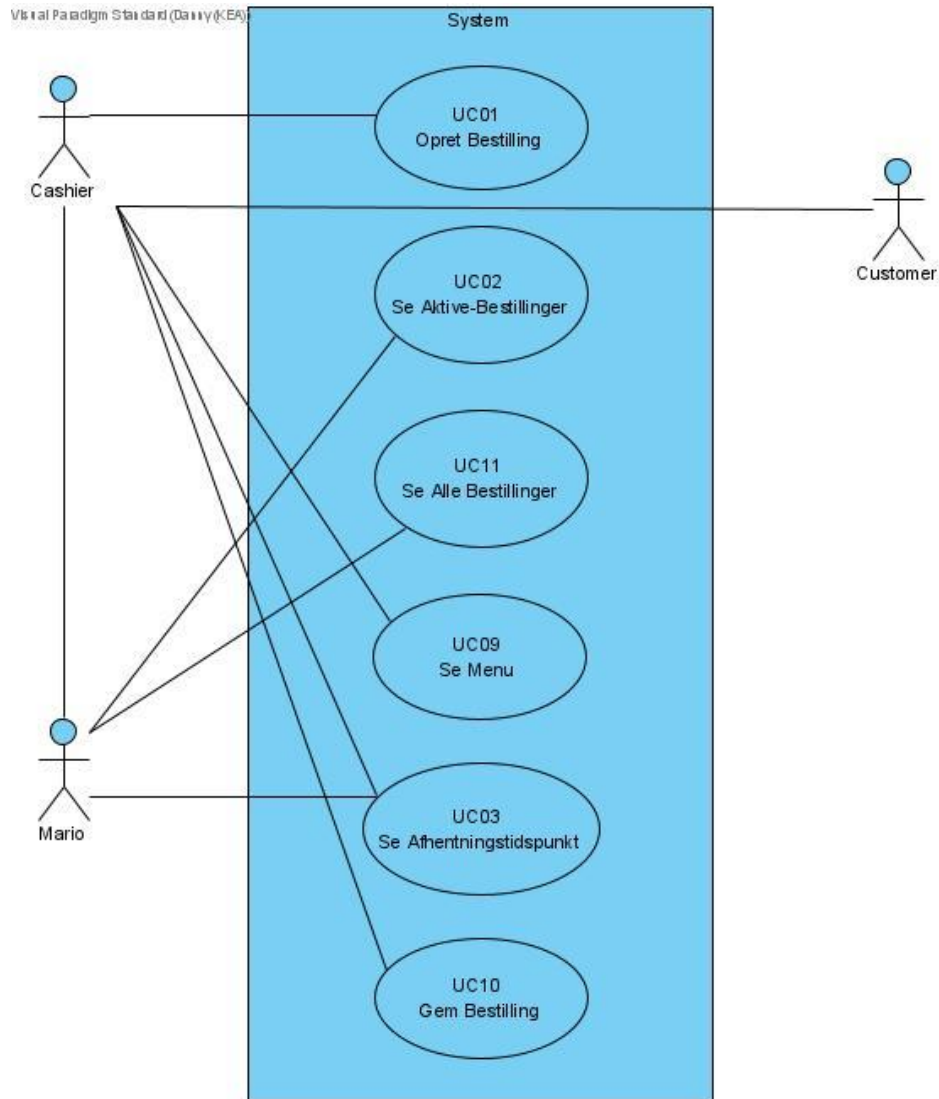
Performance

Id	Requirement	Comment
PR001	Hurtig responstid	Da der ikke er nogen direkte kommunikation med en server, afhænger responstiden af kundens computer.
PR002	...	

Supportability

Id	Requirement	Comment
SR001		
SR002	...	

Use Cases



Use Case Section	Comment
Use Case Name	Create Order
Scope	Bestillings manager applikation
Level	Subfunction
Primary Actor	Cashier (Alfonso)
Stakeholders and Interests	<ul style="list-style-type: none"> - Mario: Ønsker at se bestillingerne. Bestillingerne skal oprettes, før de kan ses. - Alfonso: Ønsker at have en glidende overgang fra bestilling til maden bliver lavet. - Kunden: Ønsker at få deres mad, uden at vente for længe.
Preconditions	En kunde skal foretage en bestilling
Success Guarantee	Bestillingen er blevet indtastet, og Mario er i gang med at lave maden
Main Success Scenario	<ol style="list-style-type: none"> 1. Kunden foretager en bestilling 2. Alfonso giver kunden et estimeret afhentningstidspunkt 3. Alfonso indtaster ordren 4. SYSTEM opretter bestillingen 5. SYSTEM respondere
Extensions	<ol style="list-style-type: none"> 3.a . Alfonso indtaster et forkert nummer <ol style="list-style-type: none"> 1. Kunden får den forkerte pizza 2. Kunden bliver utilfreds og ønsker evt. Ny pizza. 2-*.a Kunden vil annullere sin ordre <ol style="list-style-type: none"> 1. Alfonso annullerer ordren. 5.a SYSTEM respondere at bestillingen ikke blev oprettet <ol style="list-style-type: none"> 1. Alfonso indtaster bestillingen igen og sikre sig at

	bestillingen bliver indtastet korrekt
Special Requirements	
Technology and Data Variations List	
Frequency of Occurrence	Kontinuerlig, da det bliver brugt til hver bestilling i løbet af dagen
Miscellaneous	

Use Case Section	Comment
Use Case Name	ETA
Scope	Bestillings manager application
Level	Subfunction
Primary Actor	Mario
Stakeholders and Interests	<ul style="list-style-type: none"> - Mario: vil gerne se afhentningstidspunkt så han kan lave maden til tiden. - Alfonso: Ønsker at have en glidende overgang fra bestilling til maden bliver lavet. - Kunden: Ønsker at få deres mad, uden at vente for længe.
Preconditions	Mario skal kunne se, hvad han skal tilberede.
Success Guarantee	

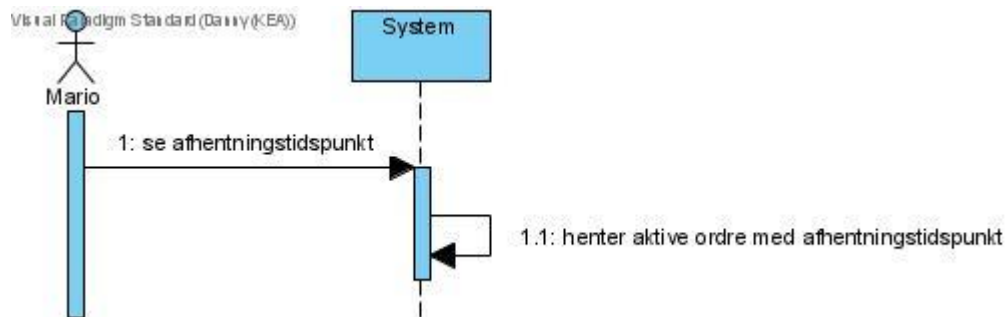
Main Success Scenario	<ol style="list-style-type: none"> 1. SYSTEM viser oversigten over bestillingerne. 2. Mario kigger på oversigten. 3. Mario laver maden. 4. Mario råber til Alfonso, når maden er klar. 5. Alfonso tager maden fra Mario. 6. Alfonso leverer maden til kunden.
Extensions	<ol style="list-style-type: none"> 1.a Der er ingen bestillinger 2.a Mario laver den forkerte bestilling
Special requirements	
Technology and Data Variations List	
Frequency of Occurrence	Kontinuerlig, da det bliver brugt til hver bestilling i løbet af dagen
Miscellaneous	

Use Case Section	Comment
Use Case Name	StoredOrder
Scope	Bestillings manager application
Level	Subfunction
Primary Actor	Mario
Stakeholders and Interests	<ul style="list-style-type: none"> - Mario: Ønsker at se bestillingerne. Bestillingerne skal oprettes, før de kan ses. - Cashier: Ønsker at have en glidende overgang fra bestilling til maden bliver lavet.

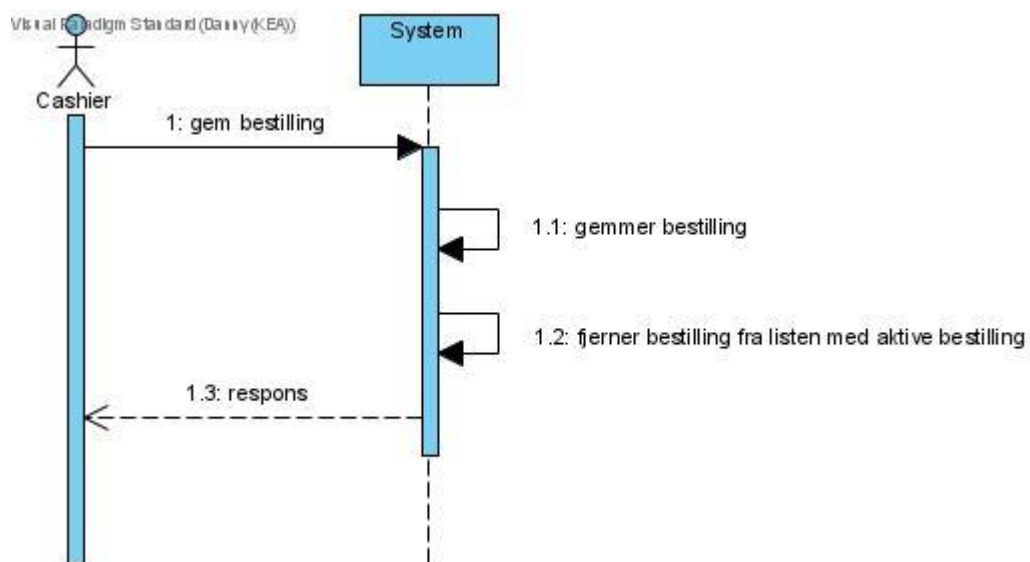
	- Customer: Ønsker at få deres mad, uden at vente for længe.
Preconditions	Mario skal kunne se hvad han skal tilberede.
Success Guarantee	What must be true on successful completion, and worth telling the reader.
Main Success Scenario	7. SYSTEM viser oversigten over bestillingerne. 8. Mario kigger på oversigten.
Extensions	1a. Der er ingen Order.
Special requirements	
Technology and Data Variations List	
Frequency of Occurrence	Kontinuerligt
Miscellaneous	

System Sequence Diagram

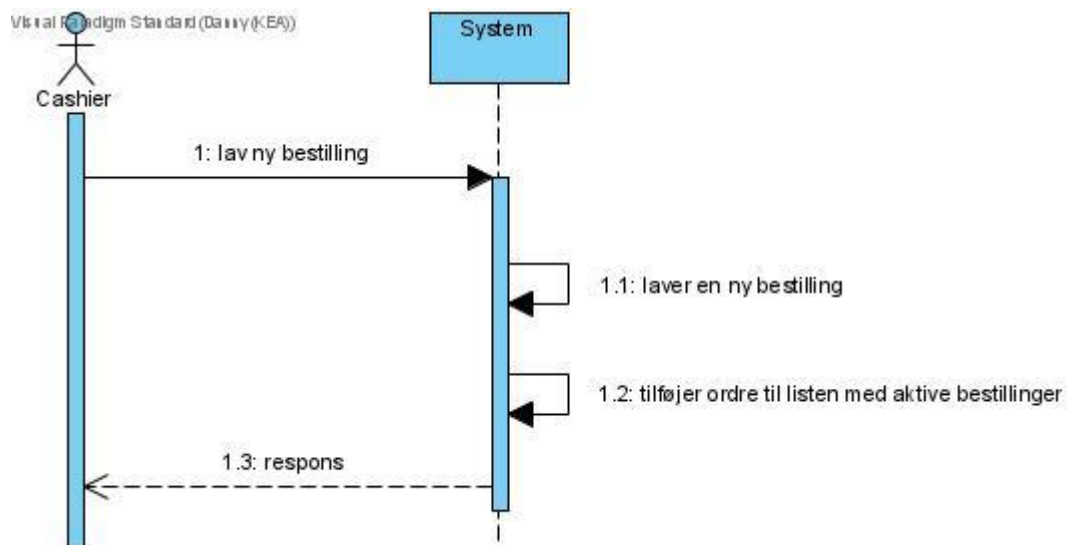
See ETA:



Store order:



Create order:

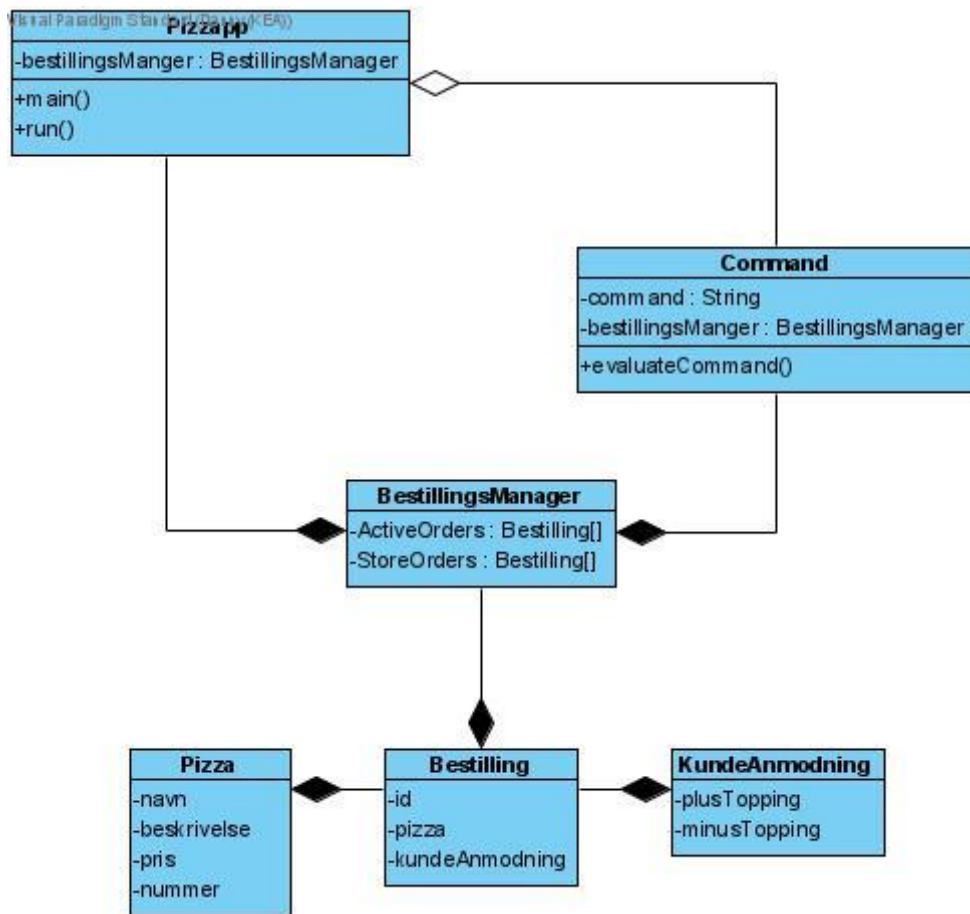


Operation Contracts

Operation	AddToActiveOrders(Order order)
Cross References	Opret Order
Preconditions	En Customer fortager en Order
Postconditions	Den nye Order er blevet tilføjet til activeOrders og bliver vist på konsolen sammen med de andre activeOrders.

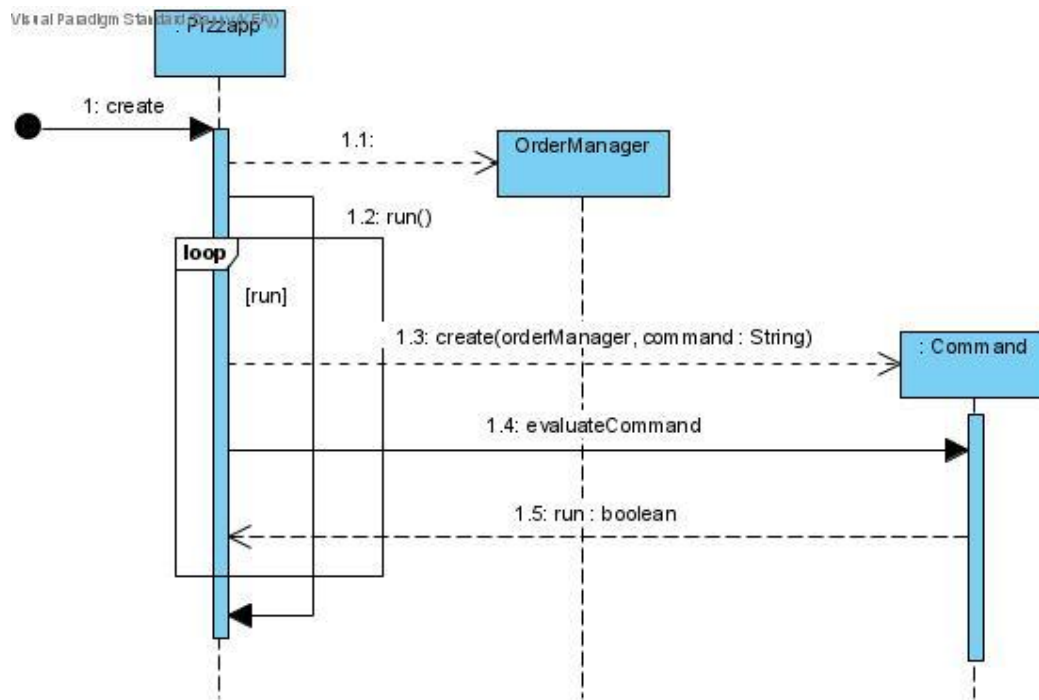
Operation	storeActiveOrders(ArrayList<Integer> indexes)
Cross References	Gem Order
Preconditions	<ul style="list-style-type: none">• Mario har færdigjort en Order
Postconditions	<ul style="list-style-type: none">• Bestillingen bliver fjernet fra <i>activeOrders</i>• Bestillingen er blevet gemt til <i>storedOrders</i>

Design Class Diagram

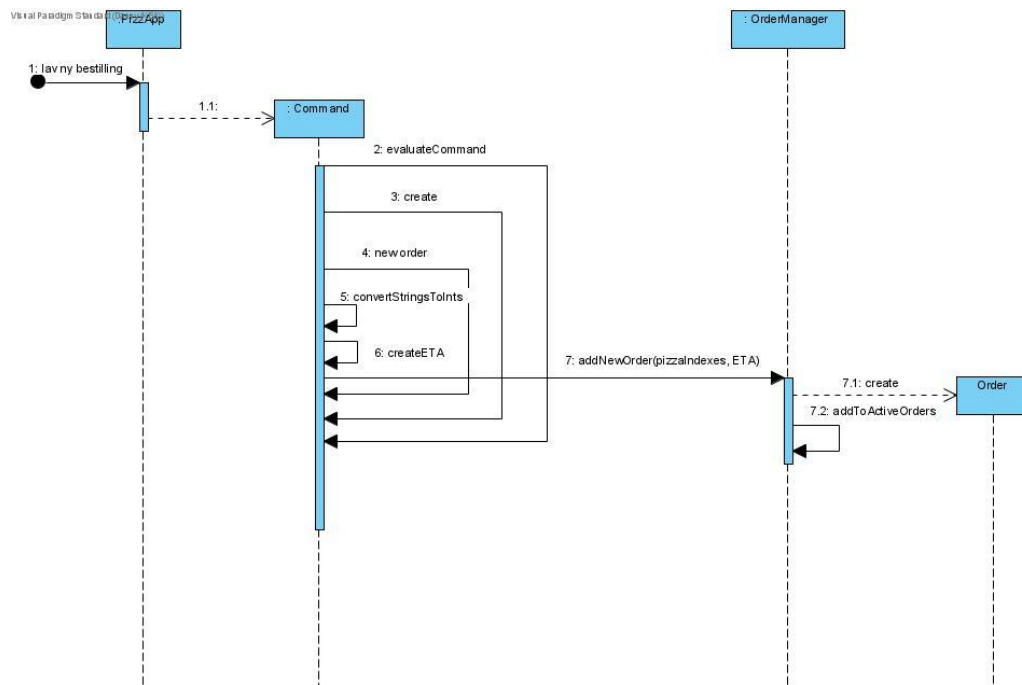


Sequence Diagram

pizzApp:

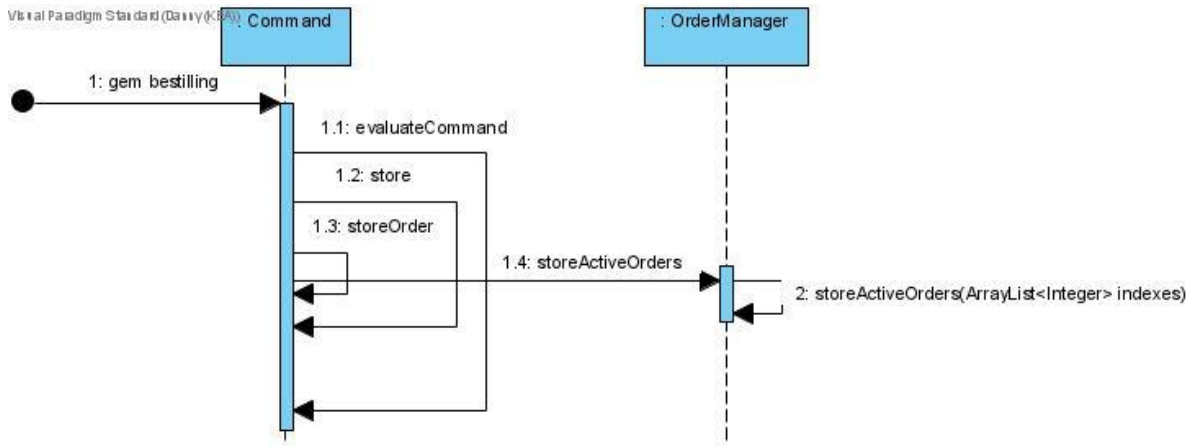


createNewOrder:



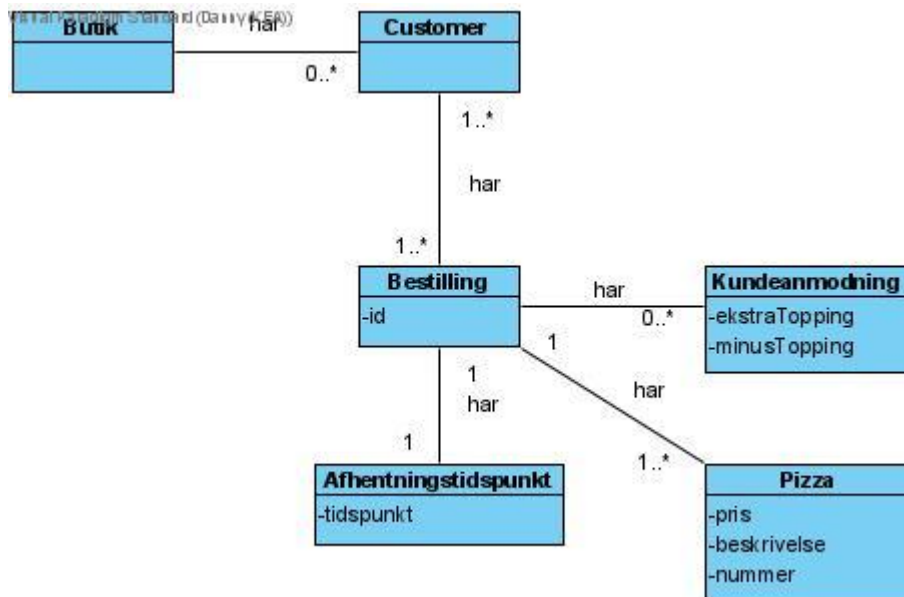
storeActiveOrder:

Visual Paradigm Standard (Daily (8.8.2019))



Domain diagram.

Domain diagram:



Glossary

Domain/Business:

Term	Definition and Information	Format	Validation Rules	Aliases
Afhentningstidspunkt	En liste over aktive bestillinger.	ArrayList		Aktive-bestillinger.
Bestilling	En klasse til at opbevare al data vedrørende en ordre.	Klasse		Order.
Kunden	En kunde i butikken.	Actor		Customer.
Pizza	Varen/produkt et der bliver solgt.			

Technical/Data:

Term	Definition and Information	Format	Validation Rules	Aliases
activeOrders	En liste over aktive bestillinger.	ArrayListe		Aktive-bestillinger.
addNewOrder	Laver en ny ordre.	Metode		
addToActiveOrders	Tilføjer ordren til en ArrayList med aktive ordre.	Metode		
Order	En klasse til at opbevare al data vedrørende en ordre.	Klasse		Bestilling.
Cashier	Marios nevø som betjener kunderne.	Actor		Alfonso.
convertStringsToInts	En metode	Metode		

	der konvertere tekst input fra brugere til integer data type og kalder andre metoder til at printe fejlmedelser vis der er et ugyldigt input.			
create	En metode der kalder andre metoder afhængig af hvilken type data der skal laves.	Metode		
createEta	En metode der beder brugeren om input ift. Hvornår ordren skal være klar til.	Metode		
Customer	En kunde i butikken.	Actor		Kunde, kunder.
ETA	En klasse der opbevare en LocalDateTime instans og har nogle metoder til at tilgå den.	Klasse		Afhentningstidspunkt, Estimated Time of Arrival.
EvaluateCommand	En metode der ud fra hvad brugeren har indtastet kalder andre metoder der kan pege programmet hen imod den funktionalitet brugeren ønsker.	Metode		

Mario	Ejer af Marios pizzabar.	Actor		Marios.
Menu	En liste over de forskellige pizzaer.	Klasse		
newOrder	En metode der kalder ConvertString sToInts for at få de indexes som OrderManager skal bruge for at tilføje pizzaerne til ordren.	Metode		
orderAction	En streg som beskriver den type action som convertString sToInts foretager, bruges ifm. Udskrivning af fejlmedelser.	Streg		
PizzApp	Main klassen der kører programmet via. <i>run</i> .	Klasse		
storeActiveOrders	En metode der tilføjer ordre fra aktiveOrders til storedOrders.	Metode		
removeActiveOrders	En metode der fjerner en ordre fra aktiveOrders, ordren gemmes ikke i denne forbindelse.	Metode		
storedOrders	En liste over gemte bestillinger.	ArrayListe		Gemte bestillinger.

storeOrder	Kalder convertString sToInts og sender de konverteret integers til orderManger hvor den tilføjer ordrene med de indexes til storedOrders og fjerner dem fra activeOrders..	Metode		
------------	--	--------	--	--