Persistens

Professionshøjskolen UCN

IT-Uddannelserne

Datamatiker

Gruppe 4

Nicolai Mosgaard Jensen

Lucas Østergaard

Sebastian Abildgaard

Mathias Sørensen

Óli Jákup Finnsson

DATO

13/10-2025

# Resume

# 

# Indholdsfortegnelse

Contents

[Resume 1](#_Toc211426805)

[Indholdsfortegnelse 2](#_Toc211426806)

[Introduktion 2](#_Toc211426807)

[Problem statement 2](#_Toc211426808)

[Fully dressed Use-case 2](#_Toc211426809)

[Domain model 4](#_Toc211426810)

[System sequence diagram 4](#_Toc211426811)

[Test cases 4](#_Toc211426812)

[Design class diagram 4](#_Toc211426813)

[SQL scripts 4](#_Toc211426814)

[Creation of tables in database 4](#_Toc211426815)

[Insertion of data into database 4](#_Toc211426816)

[Code Standard 4](#_Toc211426817)

[Code Examples 4](#_Toc211426818)

# Introduktion

## Problem statement

* Hvordan kan man, men brug af …, fremstille et tildresstillende program som opnår alle ønsker sat af Wastern Style Ltd.

# Fully dressed Use-case

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Place Order | |
| Primary Actor: | Employee | |
| Preconditions: | Medarbejderen har modtaget en ordre.  Medarbejderen har fundet kundens data i databasen | |
| Postconditions: | Ordren er færdigregistreret. | |
| Frequency: | 0 - 50 Gange om dagen. | |
| Flow of Events: | Actor - Employee | System - |
|  | 1. Medarbejderen placerer en ny ordre i systemet med Customer-type data. | 1. Systemet laver en ny ordre. |
|  | 1. Medarbejderen indsætter produktdata. | 1. Systemet søger efter ordrens indhold i Warehouse og returnere produktet og reservere det. |
|  | 1. 3 – 4 gentages. | 1. Ordren bliver tildelt Discount |
| Alternative Flow: | 1A. Customer-type eksisterer ikke i systemet.  Ræk ud til kunde.  4A. Systemet kan ikke finde produktet i Warehouse.  Det returnerer at produktet er invalidt. |  |

Use casen Place Order beskriver, hvordan en medarbejder hos Western Style Ltd. registrerer en ny ordre i systemet, efter at en kunde har placeret en bestilling via telefon eller e-mail. Formålet er at sikre en korrekt og effektiv håndtering af kunde- og produktdata, så varerne kan reserveres og klargøres til levering eller afhentning.

Når medarbejderen modtager ordren, starter processen med at finde kundens oplysninger i databasen. Hvis kunden ikke allerede eksisterer, skal medarbejderen oprette en ny kundepost. Herefter oprettes en ny ordre, og medarbejderen registrerer de ønskede produkter i systemet. Systemet søger derefter i lagerdatabasen (Warehouse) for at kontrollere, om produkterne er på lager, og reserverer dem midlertidigt, så de ikke kan bestilles af andre.

For hver vare, der tilføjes, opdaterer systemet lagerbeholdningen og sikrer, at der tages højde for forskellige lagre, herunder det mobile lager, som bruges til festivaler og markeder. Når alle produkter er registreret, vurderer systemet, om kunden opfylder betingelserne for rabat eller gratis levering — f.eks. hvis et klubkøb overstiger 1.500 DKK får man rabat, eller en privatkunde handler for mere end 2.500 DKK får man gratis levering.

Når alle oplysninger er registreret og kontrolleret, markeres ordren som færdig, og systemet genererer en bekræftelse, der kan bruges til fakturering og afsendelse af varer. Er findes to alternativ forløb i use casen. Det første (1A) opstår, hvis kunden ikke findes i systemet. I dette tilfælde skal medarbejderen kontakte kunden for at indhente de nødvendige oplysninger og eventuelt oprette kunden i databasen. Det andet alternativ flow (4A) opstår, hvis systemet ikke kan finde et produkt i lagerdatabasen. I så flad informeres medarbejderen om, at produktet er ugyldigt eller ikke på lager, hvorefter produktet må fjernes fra ordren eller erstattes af et andet.

Når alle trin er gennemført uden fejl, er ordren fuldt registreret og gemt i systemet.

# Domain model

A diagram of a computer

AI-generated content may be incorrect.

# System sequence diagram

A diagram of a diagram

AI-generated content may be incorrect.

# Test cases

Formålet med disse test cases er at validere, at use casen *Place Order* fungerer som forventet i både normal- og fejlscenarier.  
Testene dækker de flows, der er beskrevet i den *Fully Dressed Use Case*:

* **Basic Flow** (ordre oprettes korrekt)
* **Alternative Flow 1A** (kunden findes ikke)
* **Alternative Flow 4A** (produkt ikke på lager)

Systemet skal korrekt håndtere ordreoprettelse, produktreservation, rabat- og fragtregler samt fejlscenarier.

Use Case Scenarios

|  |  |  |
| --- | --- | --- |
| Scenario ID | Scenario Description | Flows |
| 1 | Successful place order | Basic Flow |
| 2 | Customer eksisterer ikke | Basic Flow + 1A |
| 3 | Produkter findes ikke i Warehouse | Basic Flow + 4A |

Scenario Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario 1  Successful place order | Basic Flow |  |  |
| Scenario 2  Customer eksisterer ikke | Basic Flow | 1A |  |
| Scenario 3  Produkter findes ikke i Warehouse | Basic Flow | 4A |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Scenario / Condition | Customer  Email | Customer  Type | Product Name | Product Type | Warehouse Quantity | Discount  Type |
| 1 | Succesfully Placed Order | V | V | V | V | V | N/A |
| 2 | Ikke nok varer på warehus | V | V | V | V | I | N/A |
| 3 | Product Type findes ikke | V | V | V | I | N/A | N/A |
| 4 | Product name findes ikke | V | V | I | N/A | N/A | N/A |
| 5 | Customer typen findes ikke | V | I | N/A | N/A | N/A | N/A |
| 6 | Email findes ikke | I | N/A | N/A | N/A | N/A | N/A |

Test Case Matrix

Disse test cases sikrer, at Place Order-processen fungerer korrekt i både normal- og fejlscenarier, samt at systemet håndterer rabat-, fragt- og lagerregler i overensstemmelse med forretningskravene fra casen Western Style Ltd.

# Design class diagram

A diagram of a computer program

AI-generated content may be incorrect.

# SQL scripts

## Creation of tables in database

## Insertion of data into database

# Code Standard

# Code Examples