

# Projektuppgift

Goldstarr-Trading

Sysm4 Grupp 1

Gustaf R, Carl M, Alex H, Jenny W, Yahya A, Samy K



# Innehåll

1. Planering
2. WBS/UML
3. Visning av app
4. Summering av projekt

# 1.

## Planering



- Scrum

Scrum

09.00 - 15.00 Mon-Fri

Daily morning scrums

Daily afternoon scrums

Backlogging

- Gruppindelning

Agila roller och indelningar.

Primära grupper:

Grupp 1: Gustaf, Alex, Carl

Grupp 2: Yahya, Samy, Jenny

- Uppgiftsindelning

Gustaf = Customerinriktat,  
data/filhantering

Carl + Alex =  
Merchandise/Restock/Supplier

Yahya + Jenny + Samy =  
CustomerOrder/Orderlist



# 1. Planering

## Prioritet Release 1

1. Godkända krav
2. Funktioner
3. Simplicity
4. Lätt hantering

## Grundkrav

- En lista över alla nuvarande kunder
- En vy för att kunna lägga till en ny order för en kund
- En lista över alla nuvarande produkter
- En vy som listar alla produkter och hur mycket som finns på lager
- En vy för att kunna hantera en leverans av produkter till lagret.

● Release 1 Deadline = 26/10 --> 2/11

Talare: Yahya A

Prio	Fredriks prio	Status	Ansvarig	Kommentar
1	All data ska spara på fil och läsas in vid programstart	Klar	Gustaf	Finns i master
2	En vy för att lägga till nya kunder	Klar	Gustaf	Finns i master
3	Utöka kundlistan med onlinekunder för den kommande online-affären	Klar	Yahya, Jenny, Samy	Finns i master
4	En lista över leverantörer	Klar	Carl, Alex	Finns i master
5	En vy för att lägga till nya leverantörer	Klar	Carl, Alex	Finns i master
6	En kunds ordrar ska sparas i systemet	Klar	Gustaf	Finns i master
7	Möjlighet för kunden att beställa flera olika produkter	Klar	Yahya, Jenny, Samy	Finns i master
	En vy för att se alla ordrar	Klar	Yahya, Jenny, Samy	Finns i master
	Sortera ordrar på kund	Klar	Gustaf/Carl/Alex	Finns i master
	Sortera ordrar på datum	Klar	Gustaf/Carl/Alex	Finns i master
	Sortera produktvyn på leverantörer eller produkter	Klar	Carl	Finns i master
	En kö för ordrar som inte kan levereras om lagret är tomt			STOP
	En vy för att skicka köade ordrar när lagret har fyllts på			STOP
	Möjlighet att ta bort leverantör utan att tappa produkter i lager	Klar	Carl	Löst i samband med implementation av vy för att se alla leverantörer
	Möjlighet att ta bort kund utan att tappa orderhistorik	Klar	Gustaf	Löst i samband med implementation av vy för att se alla ordrar
	En lista för ordrar mot leverantörer			STOP
	En vy för att lägga till nya produkter			STOP
	En mobil applikation för att kontrollera lagret hemifrån			STOP
	Rapporten		Alex, Samy, Jenny	Release 1 rapport klar, Release 2 klar

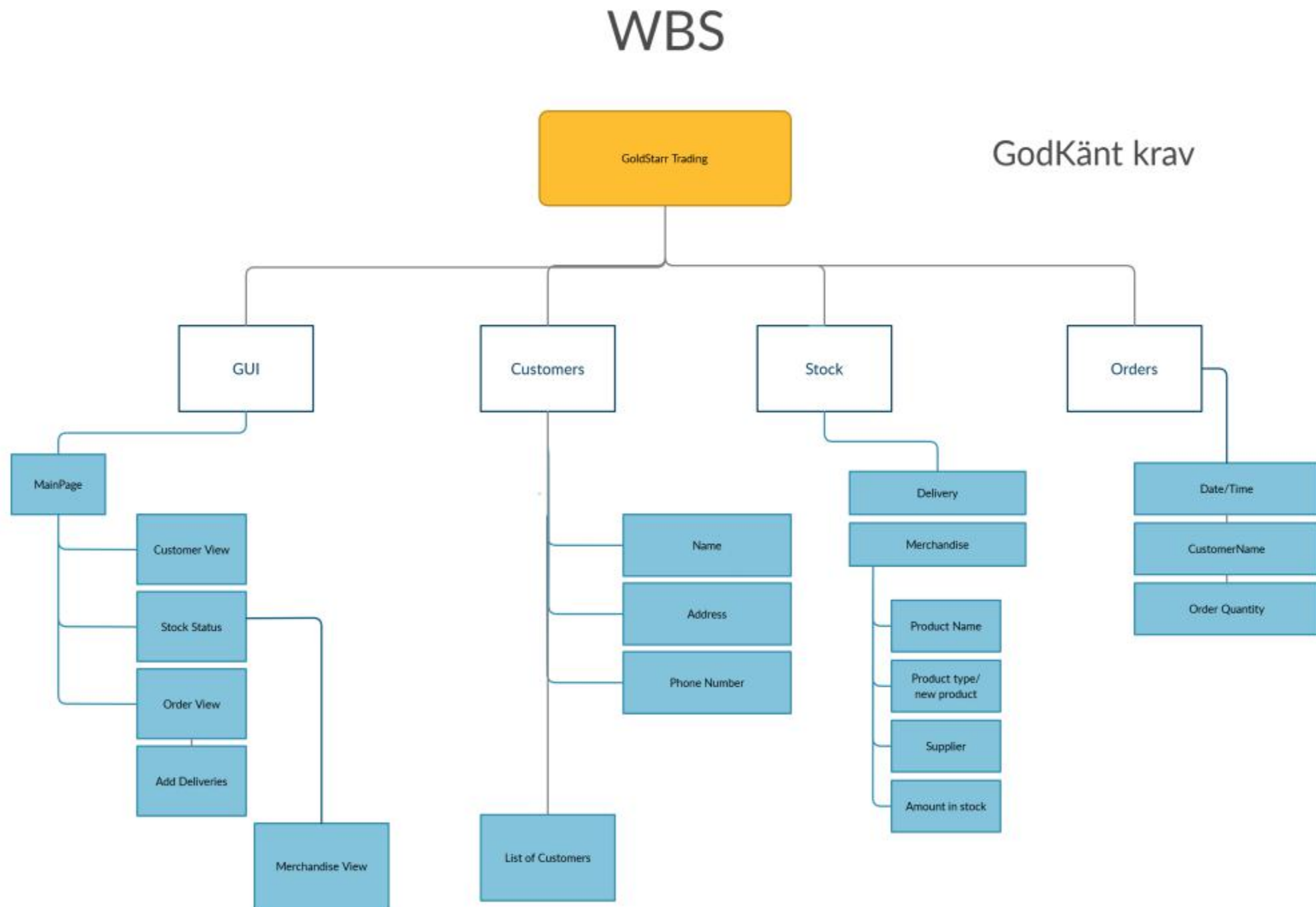
● Release 2 Deadline = 3/11 - 15/11

# 1. Planering

## Prioritet Release 2

1. Sju - Åtta krav (Fler = plus i kanten)
2. Funktioner
3. Simplicity
4. Snyggare design

Talare: Yahya A



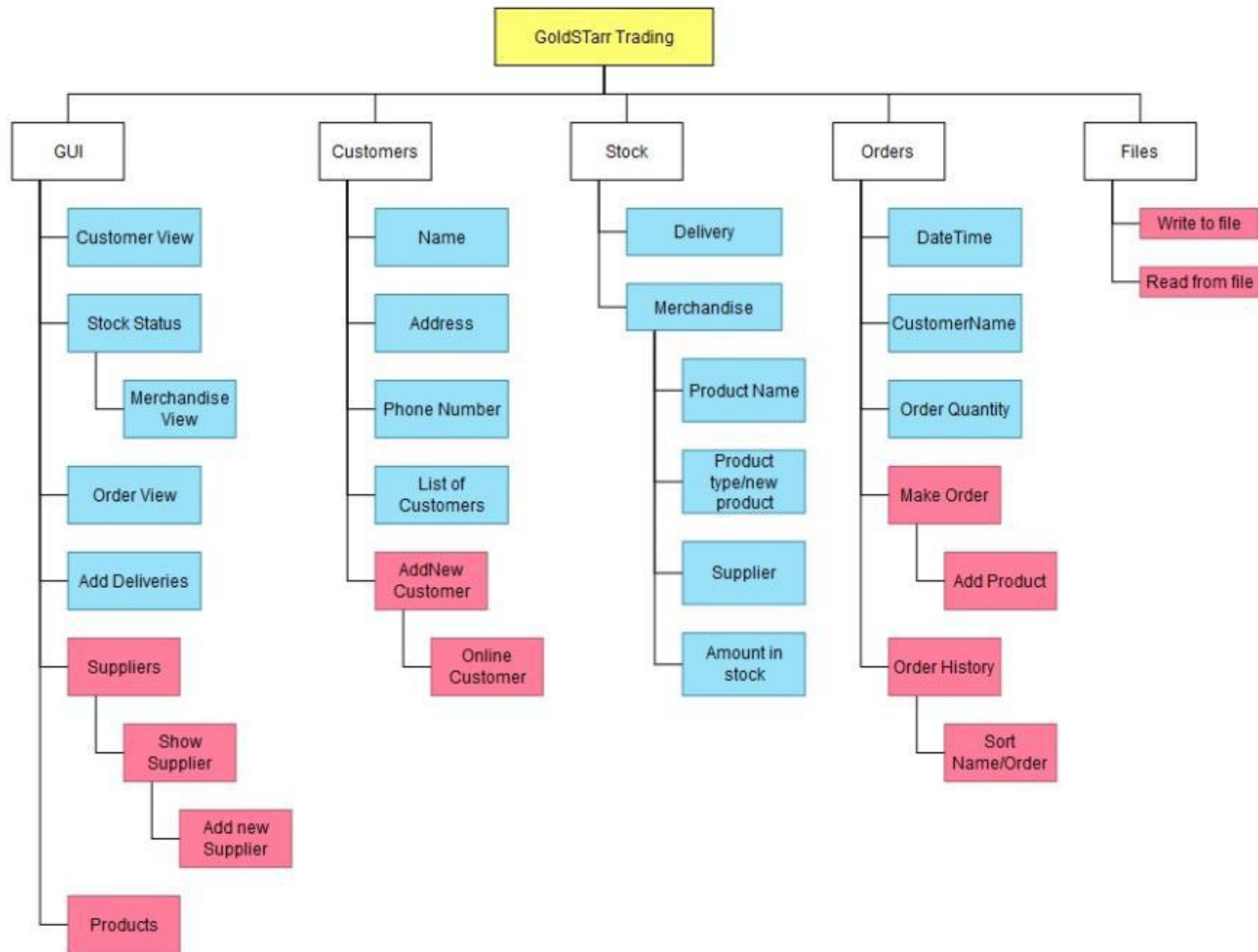


## 2. WBS Release 2

Blå = Godkänd

Röd = VÄlgodkänt

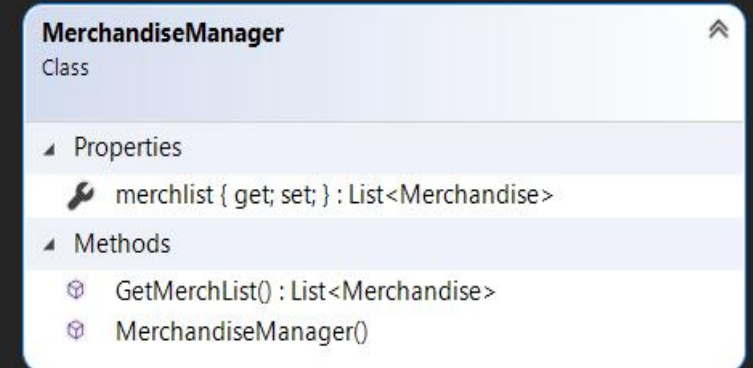
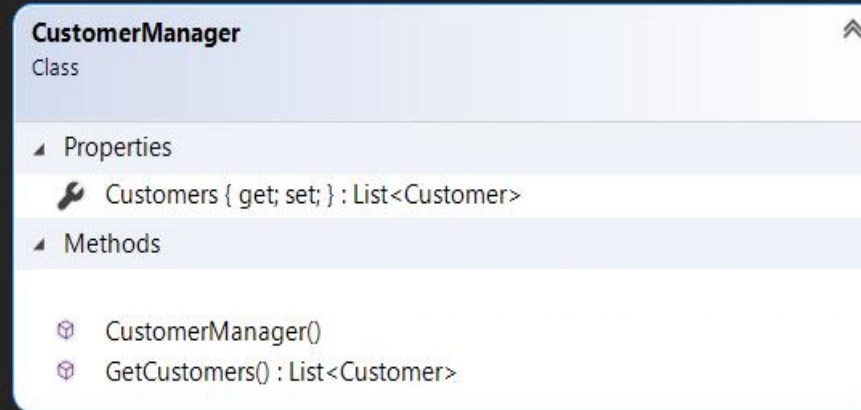
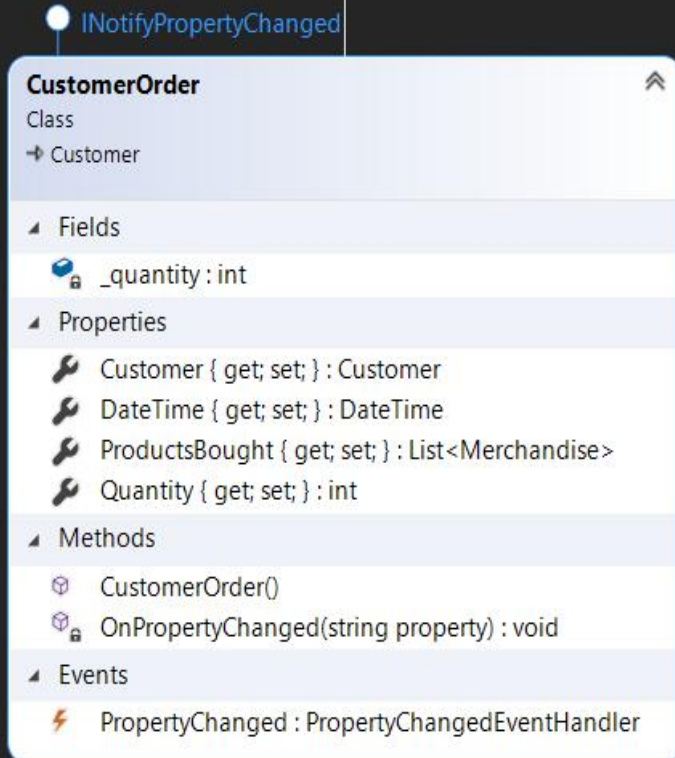
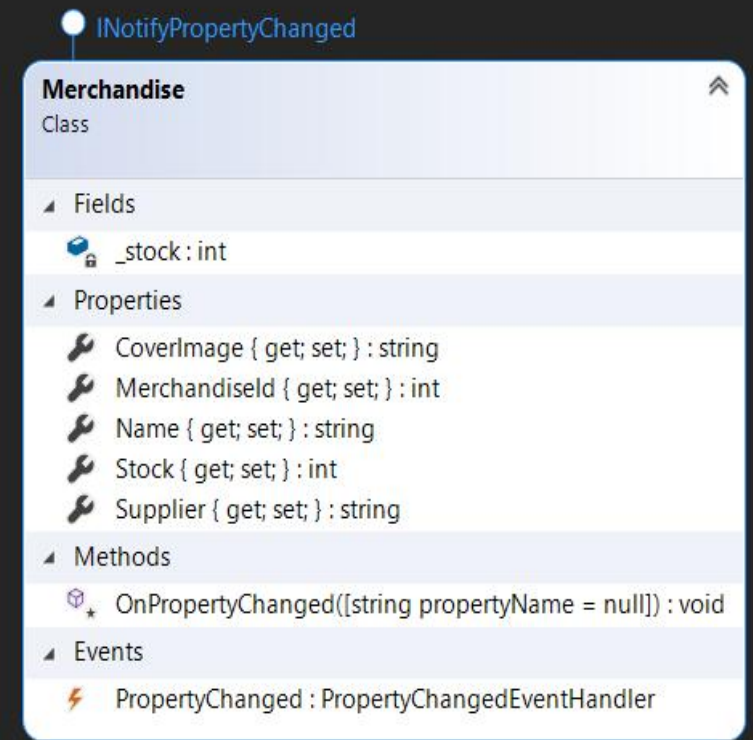
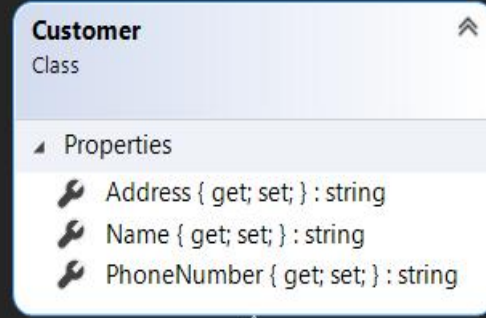
Talare: Alex H





## 2. UML Release 1

Talare: Jenny W





## CustomerType

Enum

Butikskund  
Onlinekund

# 2.

UML Release 2  
del 1

Talare: Jenny W

## Customer

Class

### Properties

- Address { get; set; } : string
- CreditCardNumber { get; set; } : string
- CustomerEmail { get; set; } : string
- DeliveryAddress { get; set; } : string
- IsOnline { get; set; } : CustomerType
- Name { get; set; } : string
- PhoneNumber { get; set; } : string

### Methods

- Customer()

## CustomerManager

Class

### Fields

- notAvailable : string

### Properties

- Customers { get; set; } : ObservableCollection<Customer>

### Methods

- AddNewUser(string name, string address, string phonenumbe
- CreateCustomers() : ObservableCollection<Customer>
- CustomerManager()

## CustomerOrder

Class

### Properties

- Customer { get; set; } : Customer
- DateTime { get; set; } : DateTime
- ProductsBoughtList { get; set; } : ObservableCollection<ProductB...

### Methods

- CustomerOrder()

## ProductBought

Class

### Properties

- Product { get; set; } : Merchandise
- ProductCurrentStock { get; set; } : int
- QuantityBought { get; set; } : int

INotifyPropertyChanged

**Merchandise**  
Class

Fields

\_stock : int

Properties

CoverImage { get; set; } : string

MerchandiseId { get; set; } : int

Name { get; set; } : string

Stock { get; set; } : int

Supplier { get; set; } : string

Methods

OnPropertyChanged([string propertyName = null]) : void

Events

PropertyChanged : PropertyChangedEventHandler

**MerchandiseManager**  
Class

Properties

merchlist { get; set; } : ObservableCollection<Merchandise>

Methods

GetMerchList() : ObservableCollection<Merchandise>

MerchandiseManager()

**Suppliers**  
Class

Properties

Email { get; set; } : string

Name { get; set; } : string

PhoneNr { get; set; } : string

**SupplierManager**  
Class

Properties

Suppliers { get; set; } : ObservableCollection<Suppliers>

Methods

AddNewSupplier(string \_name, string \_email, string \_phonenummer) : void

CreateSupplierInfo() : void

GetList() : ObservableCollection<Suppliers>

SupplierManager()

**FileManager**  
Class

Properties

\_fileName { get; set; } : string

Methods

FileManager(string fileName)

ReadFromFile<T>() : Task<T>

SaveFile<T>(T collection) : void

## 2. UML Release 2 del 2

Talare: Jenny W

3.

Visning av app



Här visar vi nu vår app :)



Talare: Gustaf R

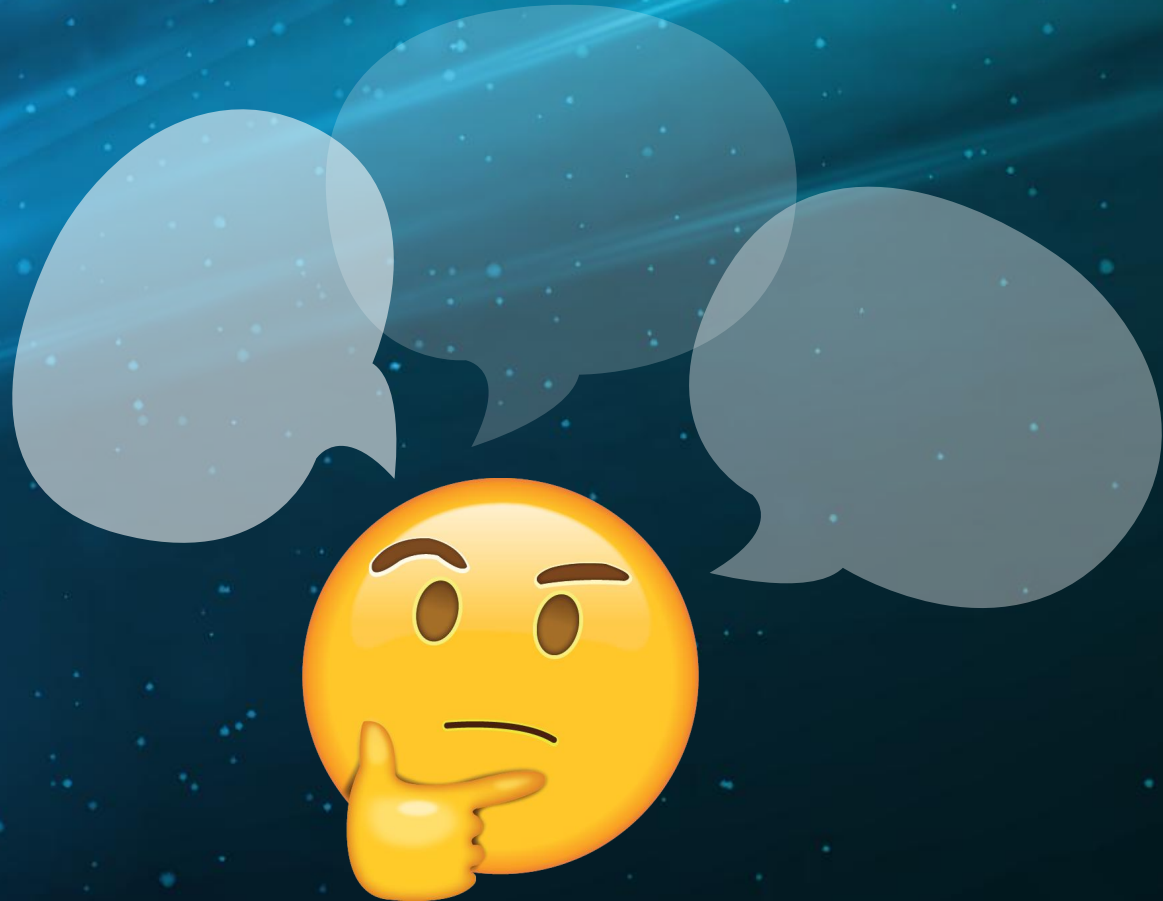


# 4.

## Sammanfattning



- Hur det har gått:
- Våra allmänna tankar
- Våra lärdomar
- Slutsats



The background is a dark blue gradient with numerous small white dots, resembling a starry night sky. Several bright blue light streaks or nebulae-like patterns sweep across the frame from the bottom left towards the top right. In the corners, there are large, low-poly geometric shapes in shades of gray and white, some appearing to be fragments of a larger structure. A small, stylized white arrow points towards the right, positioned between the words 'Slut' and 'Finito'.

**Tudeloo :)**  
**Slut**  
**Finito**