

User stories Use Case 1

- "som bruger vil jeg kunne indtaste title, price, reol, hylde og

- "som bruger vil jeg kunne gemme varen og få en bekræftelse"

- "som bruger vil jeg kunne importere en vare fra CSV-fil" - "som bruger vil jeg kunne angive placering" - "som bruger vil jeg kunne få besked ved manglende felter input/ugyldig input"

User story 1: "som bruger vil jeg kunne indtaste title, price, reol, hylde og beskrivelse" Design inputfelter i UI til title, price, reol og hylde
Design knap til at gemme data
Vis bekræftelsesbesked i UI • Implementér front end validering (f.eks. obligatoriske

UC1US1: Design

inputfelter i UI til title,

price og reol

UC1US1: Design knap til

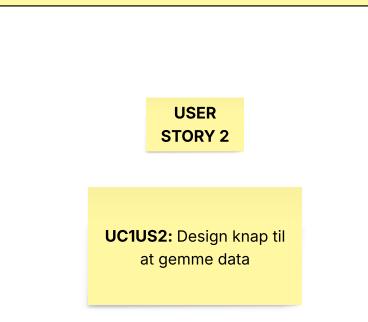
at gemme data

UC1US1: Vis bekræftelsesbesked i UI få en bekræftelse" Design knap til at gemme data Implementér logik til at gemme varer til fil Vis bekræftelsesbesked i UI Opret backend-endpoint til at modtage og gemme data Forbind frontend til backend Skriv unittest for filskrivning funktion

User story 2: "som bruger vil jeg kunne gemme varen og

User story 3: "som bruger vil jeg kunne importere en vare fra CSV-fil" • Design knap(per) til at importere fra CSV Returnér bekræftelse til frontend Implementér logik til at indlæse data fra fil Vis bekræftelsesbesked i UI Skriv unittest for indlæsnings funktion





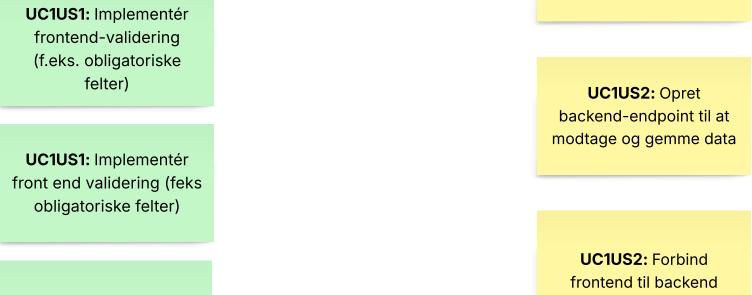
UC1US2: Skriv unittest for filskrivning funktion





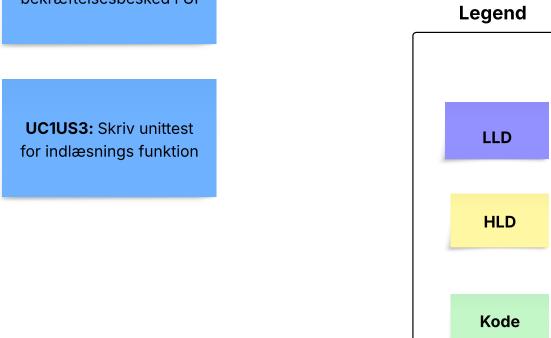


logik til at indlæse data





UC1US3: Skriv unittest



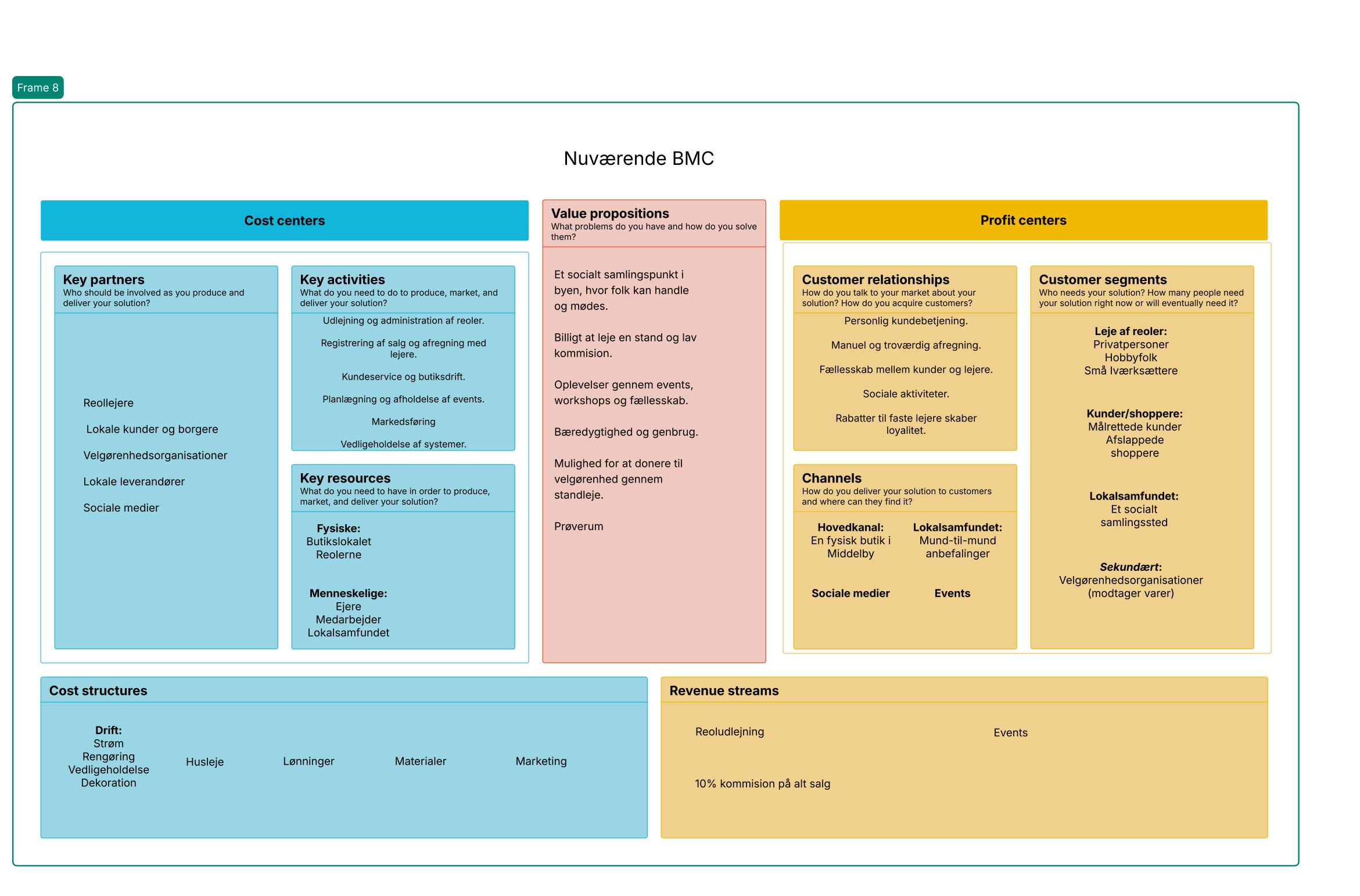
Test

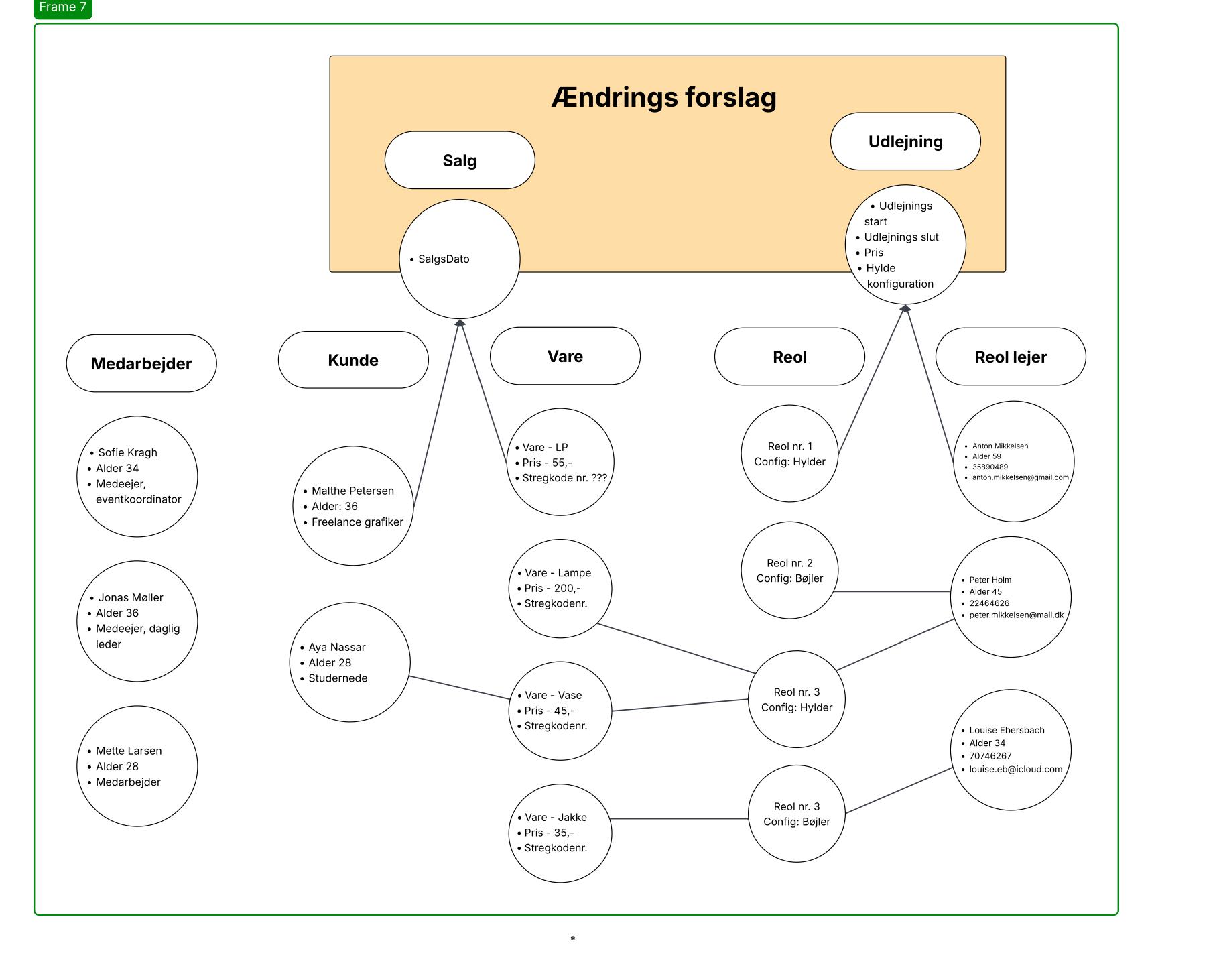
UC1US1: Forbind frontend til backend UC1US1: Test oprettelse

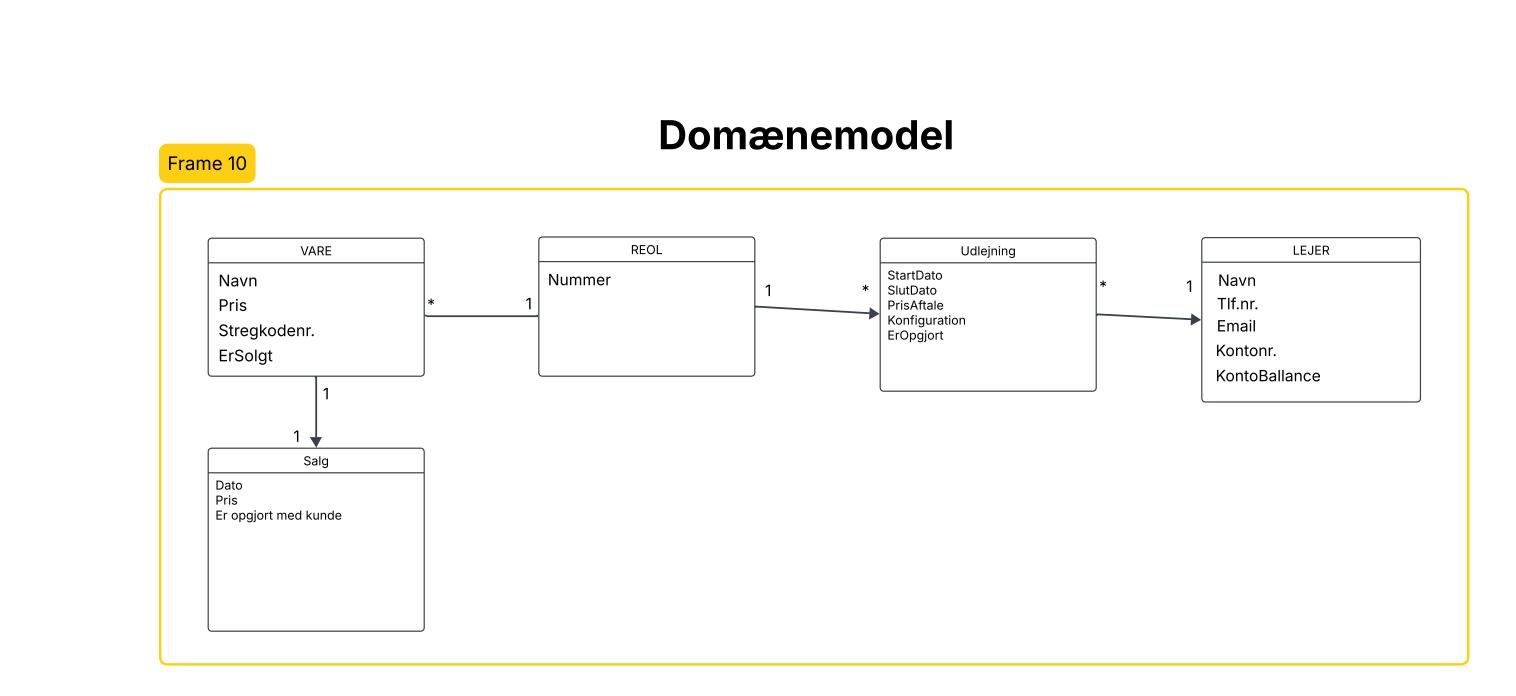
med forskellige inputs

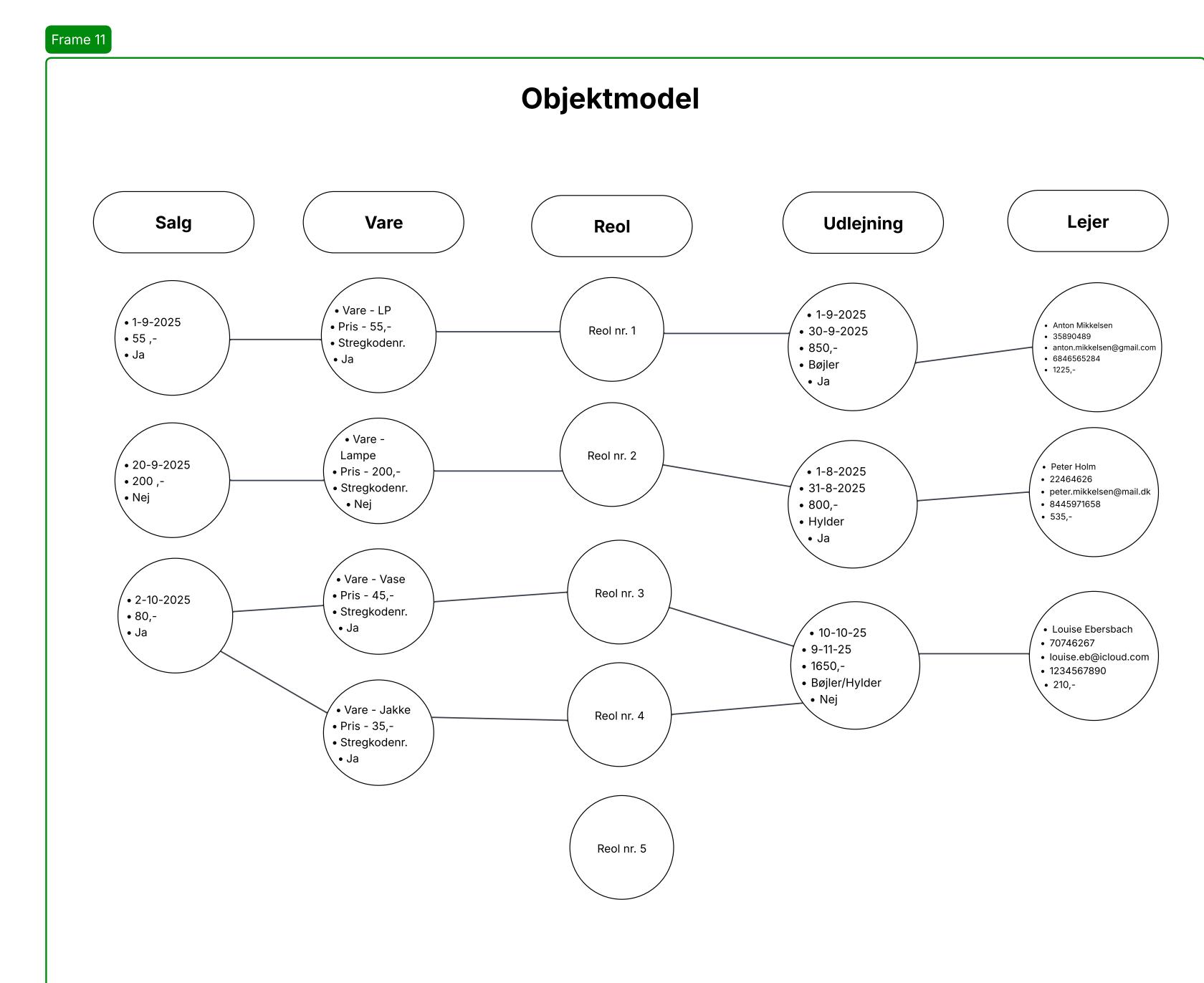
UC1US1: Opret Dummy Data til udviklings brug

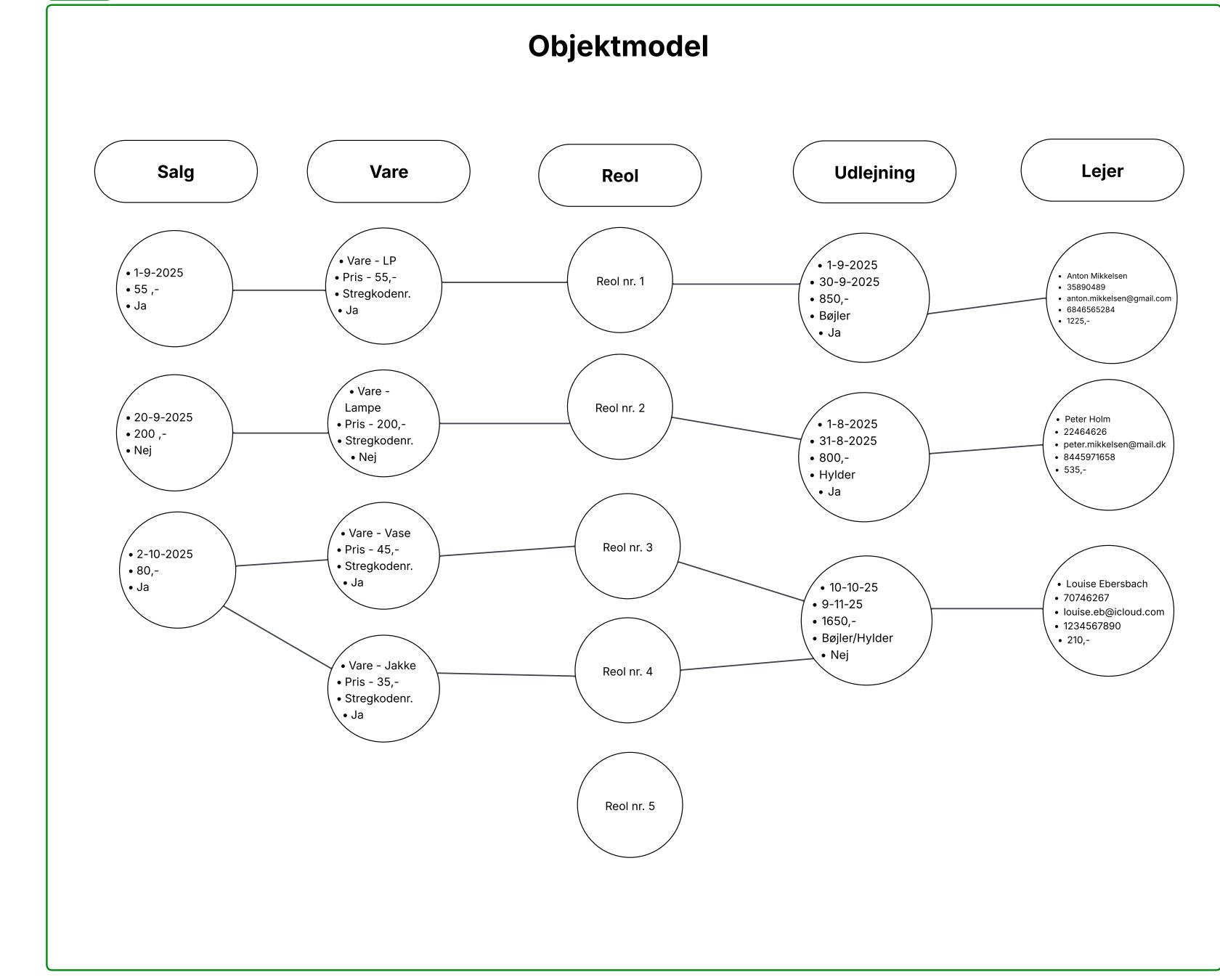
	Back Log Items	To do	In prog	gress		Done		Testing	Done Done
SPRINT 1									
	Rapportskrivning		Research bedst passende datatyper SQL	Research bedst passende datatyper C#	GIT INIT	UC1US1: Opret Database schema for Item	WPF OPSÆTNING		
	Kvalitets sikring af HLD				UC1US1: Opret Repository for ShelfUnit	UC1US1: Opret Repository for Tenant			
					UC1US1: Opret Model for Tenant				
					UC1US1: Opret Model for Sale				
					UC1US1: Opret Database schema for Sale				
					UC1US1: Opret Repository for Sale				
					uclus1: Opret Database schema for Tenant				
SPRINT 2					UC1US1: Opret Model for ShelfUnit		UC1US1: Opret Database		
SPI					UC1US1: Opret Model for Rental	Repository for Item	schema for Rental		
					uclus1: Opret Database schema for ShelfUnit				
					UC1US1: Opret Model for Item				
					UC1US1: Opret Repository for Rental				
					DATABASE CONNECTION WPF BASIC				
					OBJECT MODEL				
					ВМС				
					DOMAIN MODEL				
SPRINT 3									
SPRINT 4									

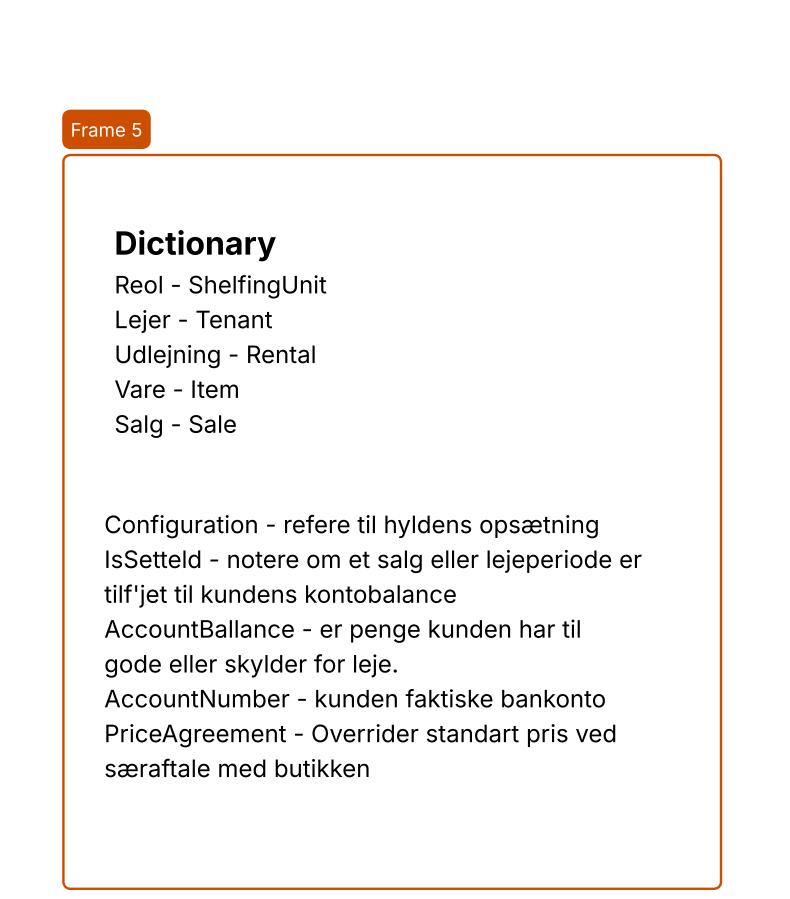


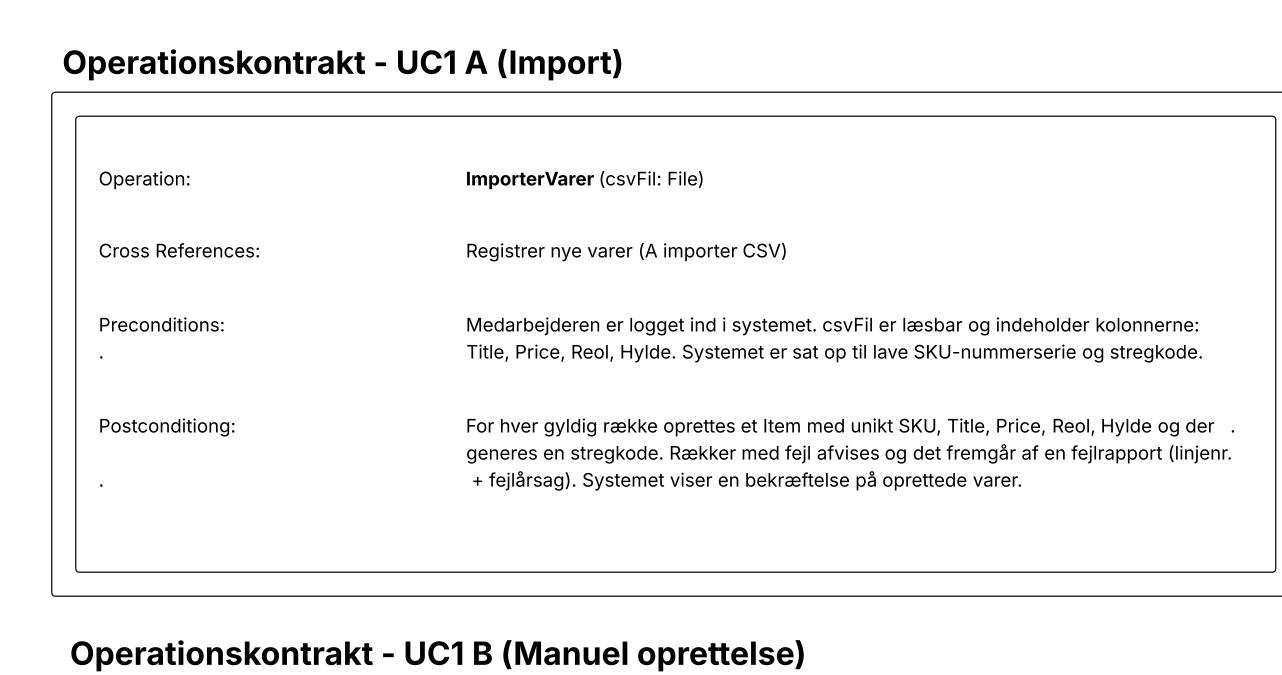












Hovedscenarie:

1. Aktør vælger "Registrer varer".

2. System viser to valg: Importér CSV eller Ny vare.

A. Importér CSV

3) Aktør vælger CSV-fil (krav: Title, Price, Reol, Hylde).

4) System viser forhåndsvisning og validering af rækker.

5) Aktør klikker "Importér".

6) System opretter varer, tildeler unikke SKU'er og genererer regkoder.

stregkoder.

7) System viser bekræftelse (antal oprettet + fejloversigt).

8) Aktør vælger "Print labels" (valgfrit) → system udskriver / gemmer som PDF.

B. Opret enkelt vare (manuel)

3) Aktør vælger "Ny vare".

4) System viser felter: Title, Price, Reol, Hylde.

5) Aktør indtaster data og vælger "Gem".

6) System opretter varen, tildeler SKU og genererer stregkode

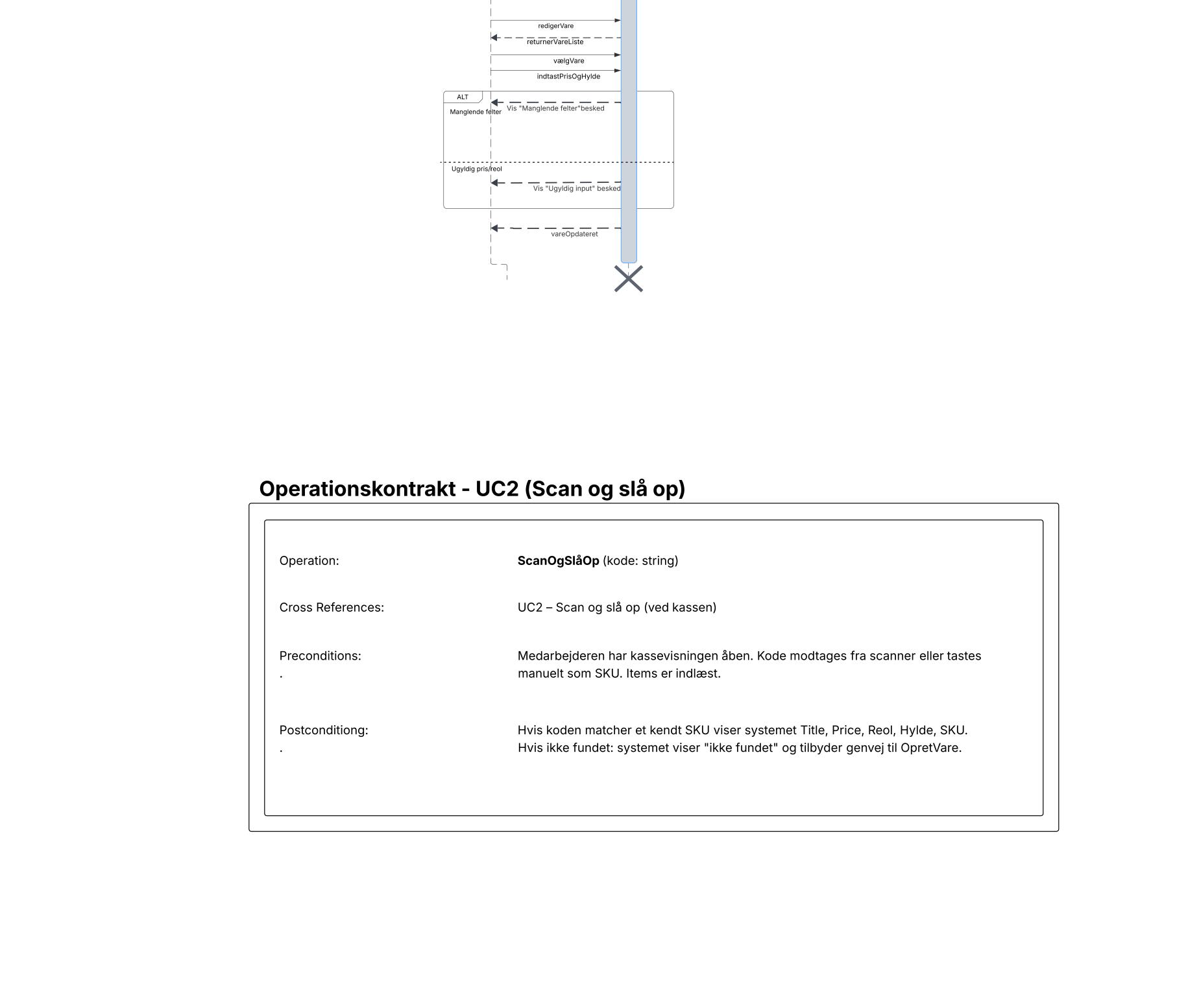
7) Aktør vælger "Print label" (valgfrit) → system udskriver / gemmer som PDF.

Alternative scenarier:
A-4a: Manglende kolonner/ugyldige værdier → Systemet markerer fejl pr. linje; aktør kan importere kun gyldige rækker.
B-2a: Aktør efterlader felt tomt/ugyldigt → Systemet viser valideringsfejl, gem afvises indtil rettet.

"som bruger vil jeg kunne importere en vare fra CSV-fil"
"som bruger vil jeg kunne angive placering"
"som bruger vil jeg kunne gemme varen og få en bekræftelse"
"som bruger vil jeg kunne få besked ved manglende felter input/ugyldig input"

VARER - Use Case 1

Operation:	OpretVare (title: string, price: decimal, reol: string, hylde: string)
Cross References:	UC1 – Registrer nye varer (B. Opret enkelt vare)
Preconditions:	Medarbejderen er logget ind i systemet. Felterne er udfylde og valide (fx Price: 40kr.) SKU og stregkodegenerator er tilgængelige.
Postconditiong:	Et nyt Item er oprettet (instance creation) med unikt SKU, Title, Price, Reol, Hylde sa genereret stregkode. Varen fremgår nu i den opdaterede liste. Klar til print af label.



vare og betale for reol leje. Medarbbejder kan finde lejerns kontobalance

3. Aktør indtaster søgeord og trykker "Søg".
4. System viser resultater: **Titel, Pris, Reol-Hylde**. **Alternative scenarier:**4a) Ingen resultater → System viser "Ingen match"

Use Case 4 - Søgefunktion for kunder

under deres kundekort på computeren og afregne med dem ved kassen,

eller vælge andre afregnings metoder. Hvis der er nogle spørgsmål t afregningen kan alle salgsposter og udlejninger findes frem i systemet.

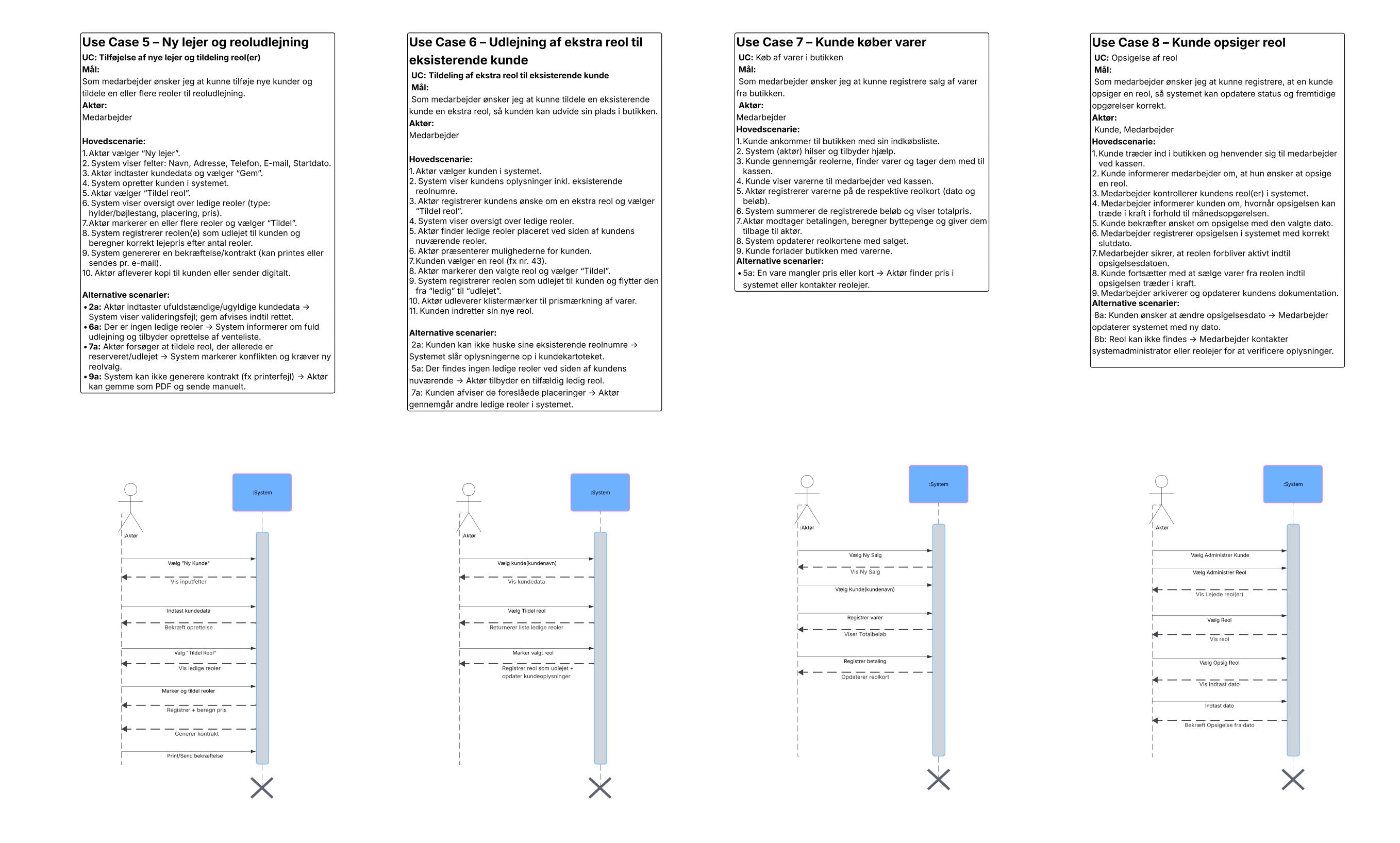
vælgReolNummerFraListe

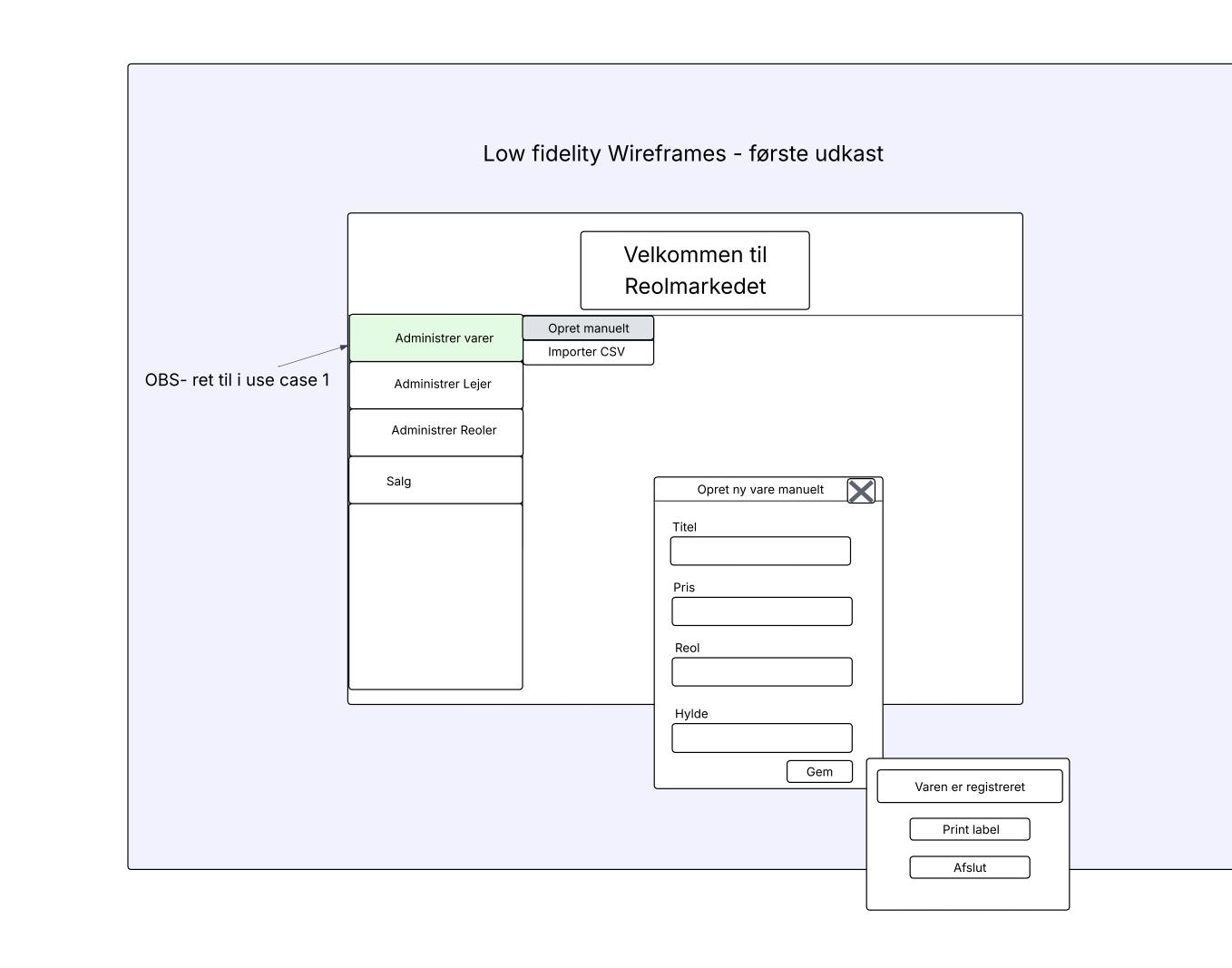
visVareListe(navn, pris, hylde, SKU)

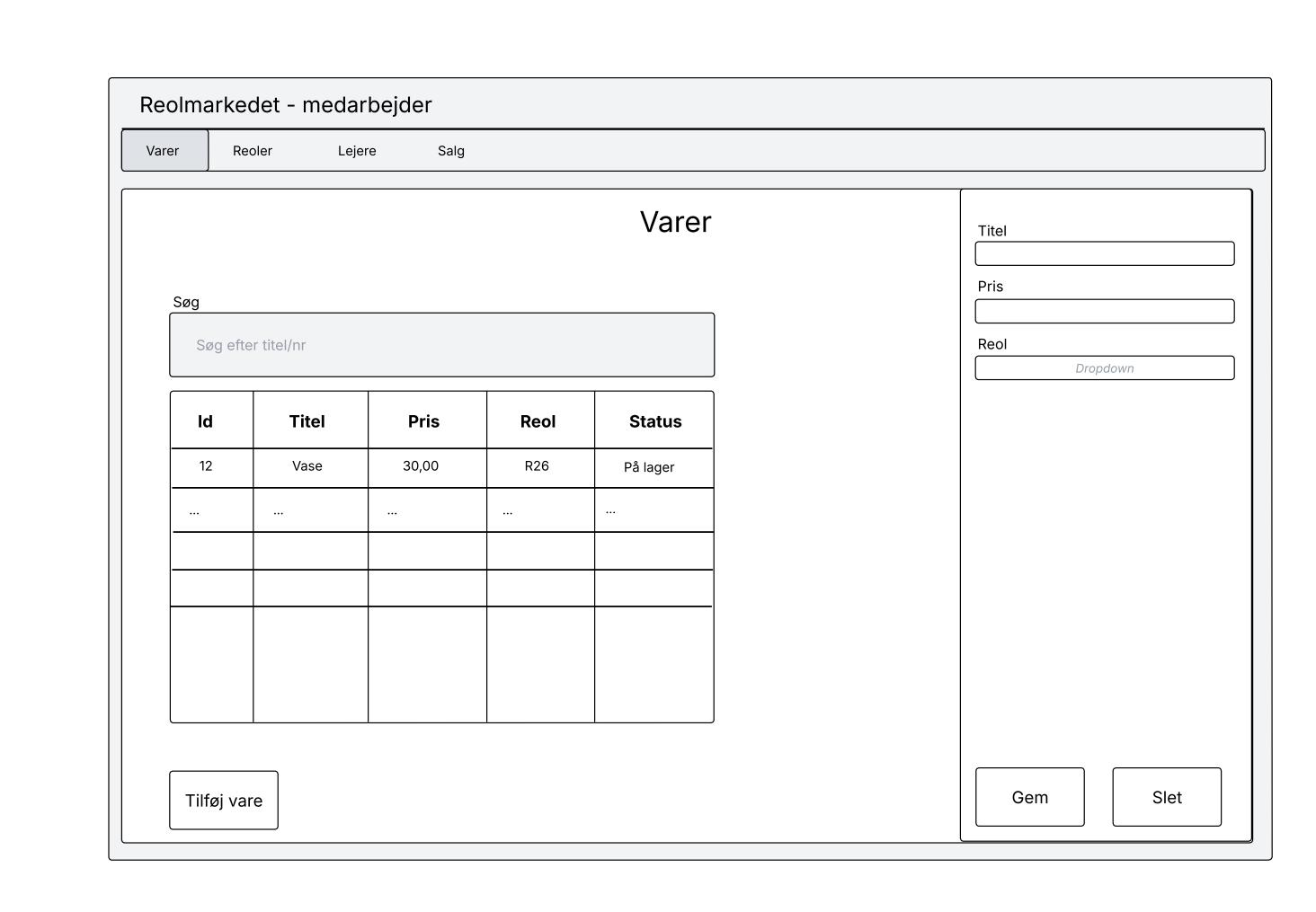
markerVare

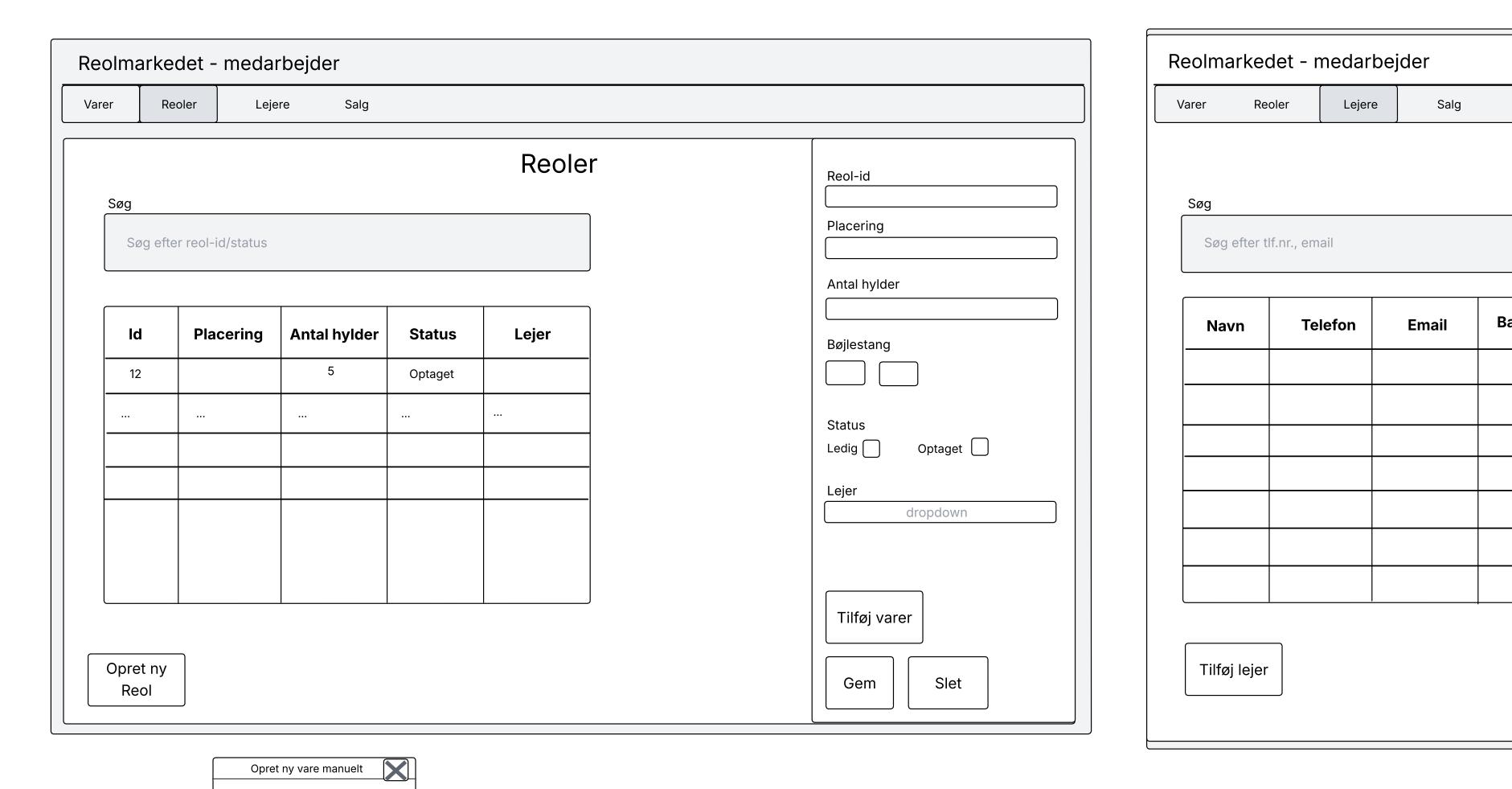
vælgFlytVare

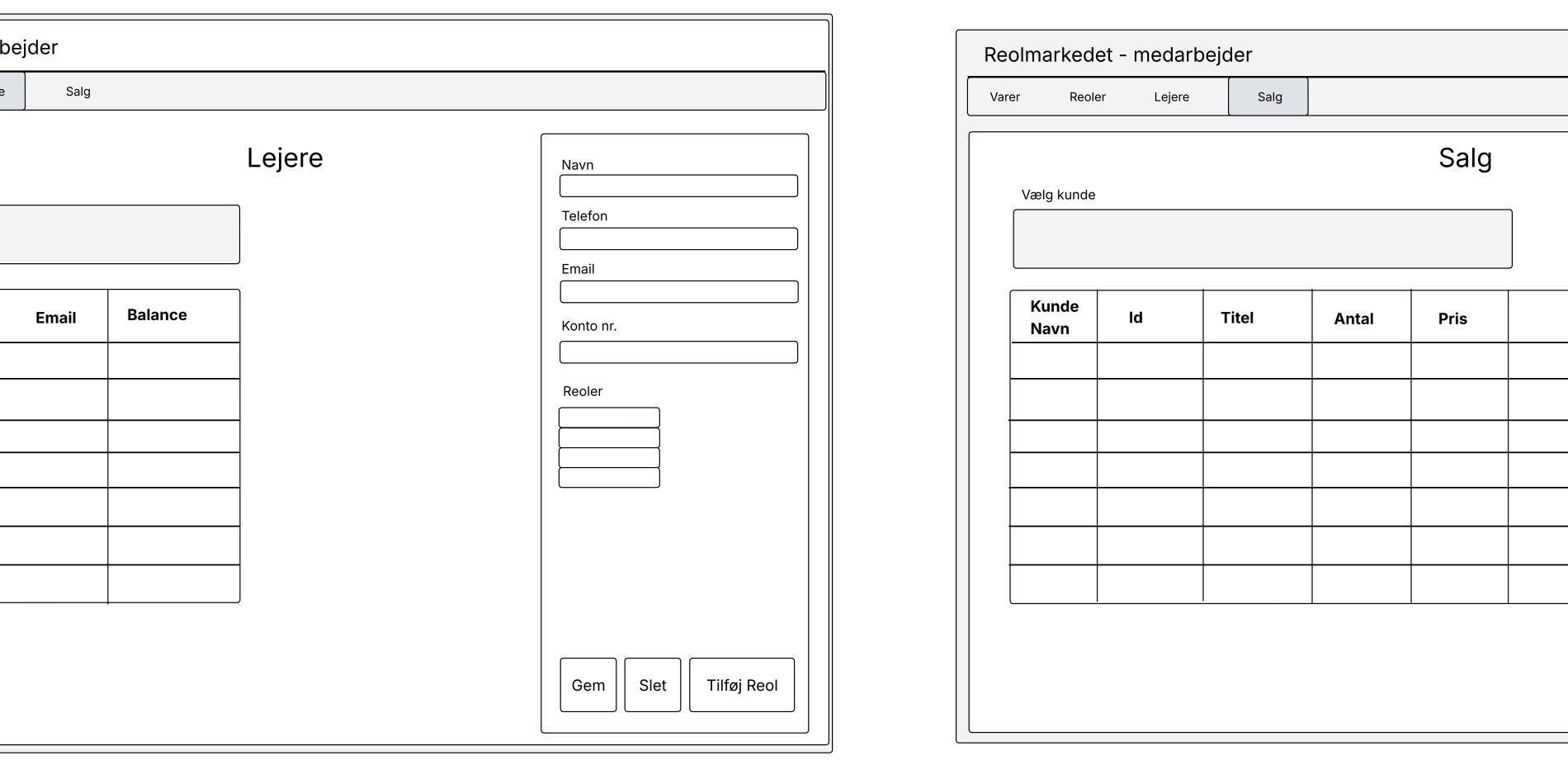
returnerNyInfo(ReoInr, Hyldenr)

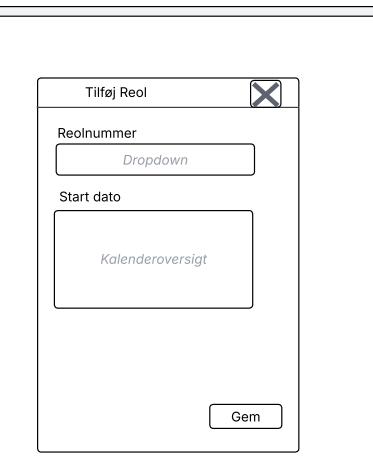






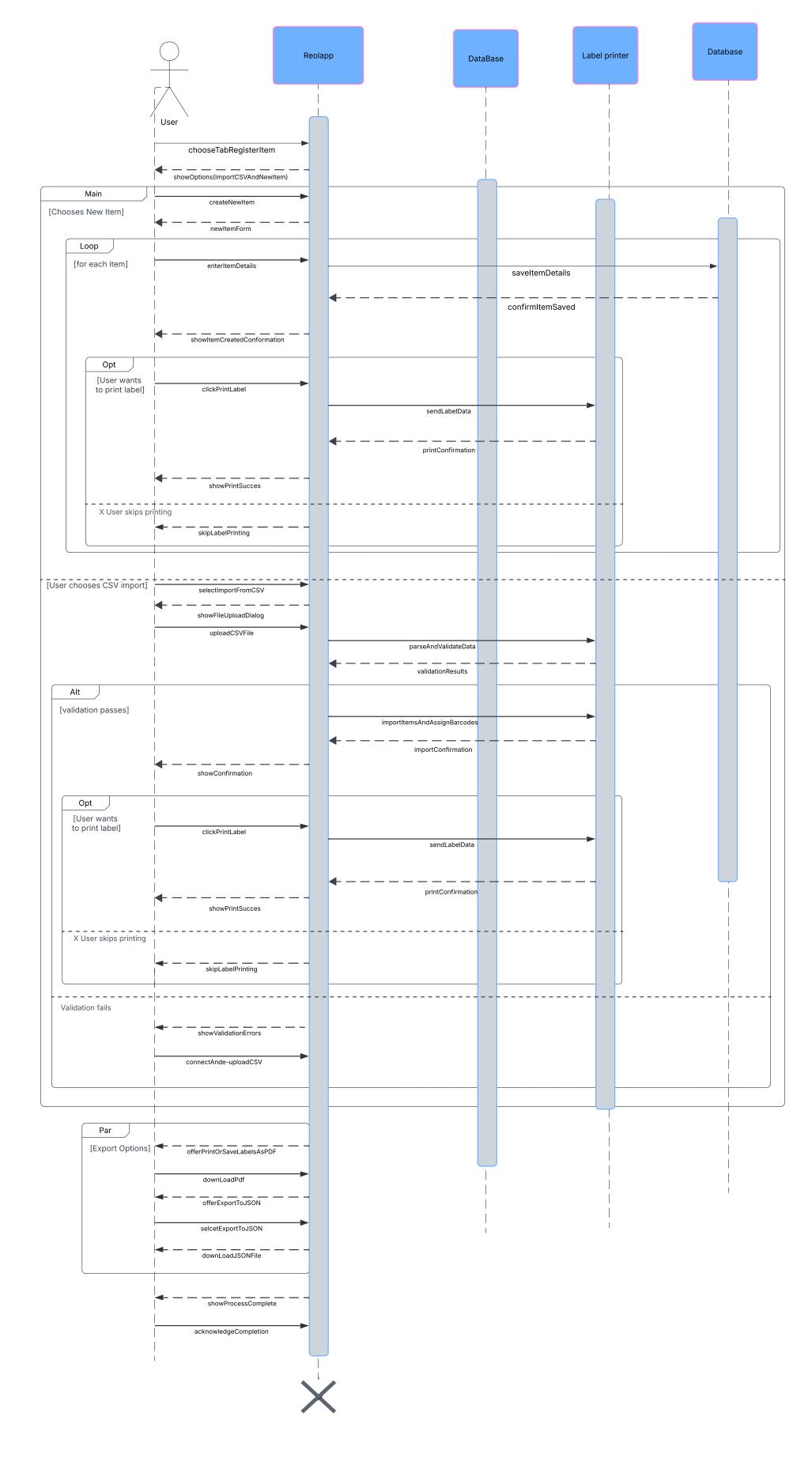




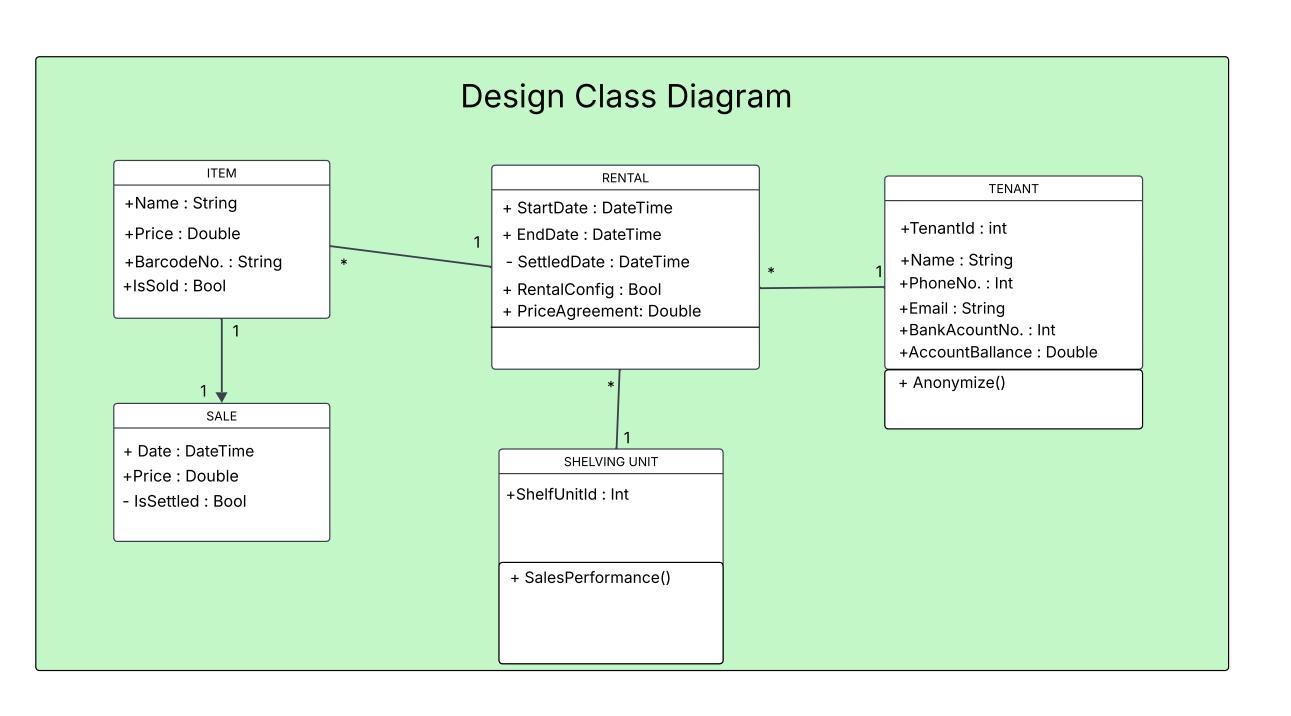


Sekvensdiagram

ITEMS - Use Case 1



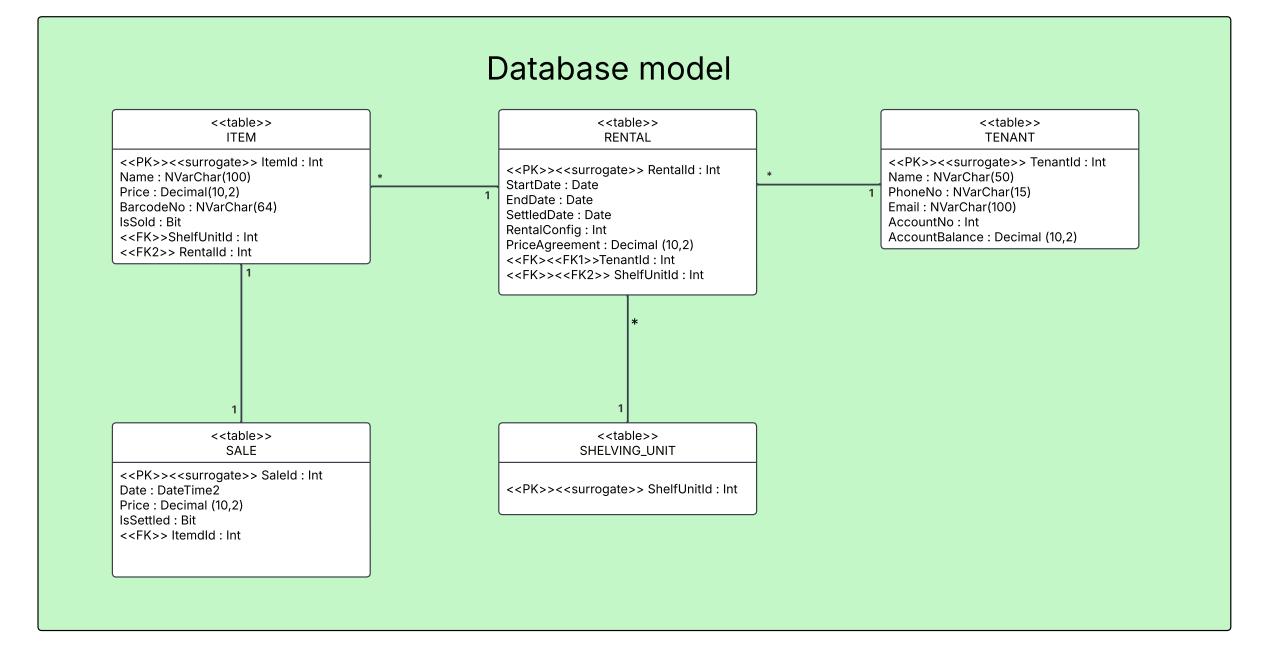
+ CreateTenant() + UpdateTenant() + DeleteTenant()



Frame 4

ITEM(ItemId, Name, Price, Barcode, IsSold, RentalId)
SALE(SaleId, ItemId, Date, Price, IsSetteId)
SHELFUNIT(ShelfUnitId)
RENTAL(RentalId, TenantId, ShelfUnitID, StartDate, EndDate, SettledDate, RentalConfig, PriceAgreement)
TENANT(TenantId, Name, PhoneNumber, Email, AccountNumber, AcountBallance)

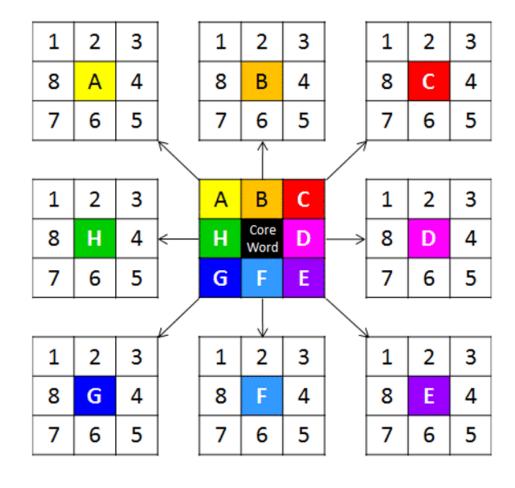
SALE SHELFUNIT RENTAL TENANT <<table>> <<table>> <<table>> <<table>> <<table>> SaleId: Int ItemId: Int ShelfUnitd: Int Rentalld: Int Tenantld: Int Name: NVarChar(100) Price: decimal (10,2) Name: NVarChar(50) Tenantld: int Price: decimal (10,2) Date: Date pk (ShelfUnitd) PhoneNumber: NVarChar(15) ShelfUnitId: int Barcode: NVarChar(64) Issetteld: bool Email: NVarChar(100) StartDate: Date IsSold: Bool ItemId: int EndDate: Date AccountNumber: int Rentalld: int IsSetteldDate: Date AccountBallance: Decimal(10,2) pk (Saleld) RentalConfig: Int fk ITEM(ItemId) pk (ItemId) pk (Tenantld) PriceAgreement:decimal fk RENTAL(Rentalld) (10,2) pk (Rentalld) fk TENANT(Tenantld) fk SHELFUNIT(ShelfUnitd)



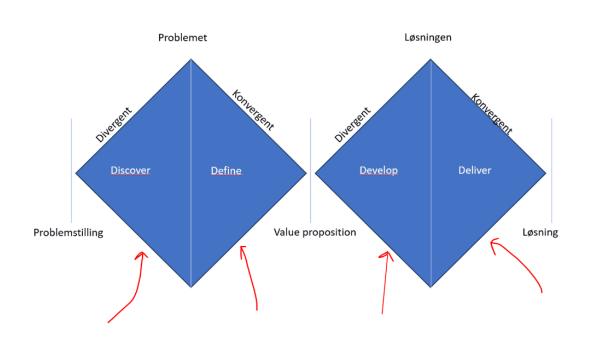
Frame

```
Data Typer SQL:

Id int
Phone # varchar
Email varchar
Desc varchar
Name varchar
Ssn varchar
Price decimal, money, smallmoney
ShipDate datetime
Sex bit
Discont bit
Quantity int
ZipCode varchar
```

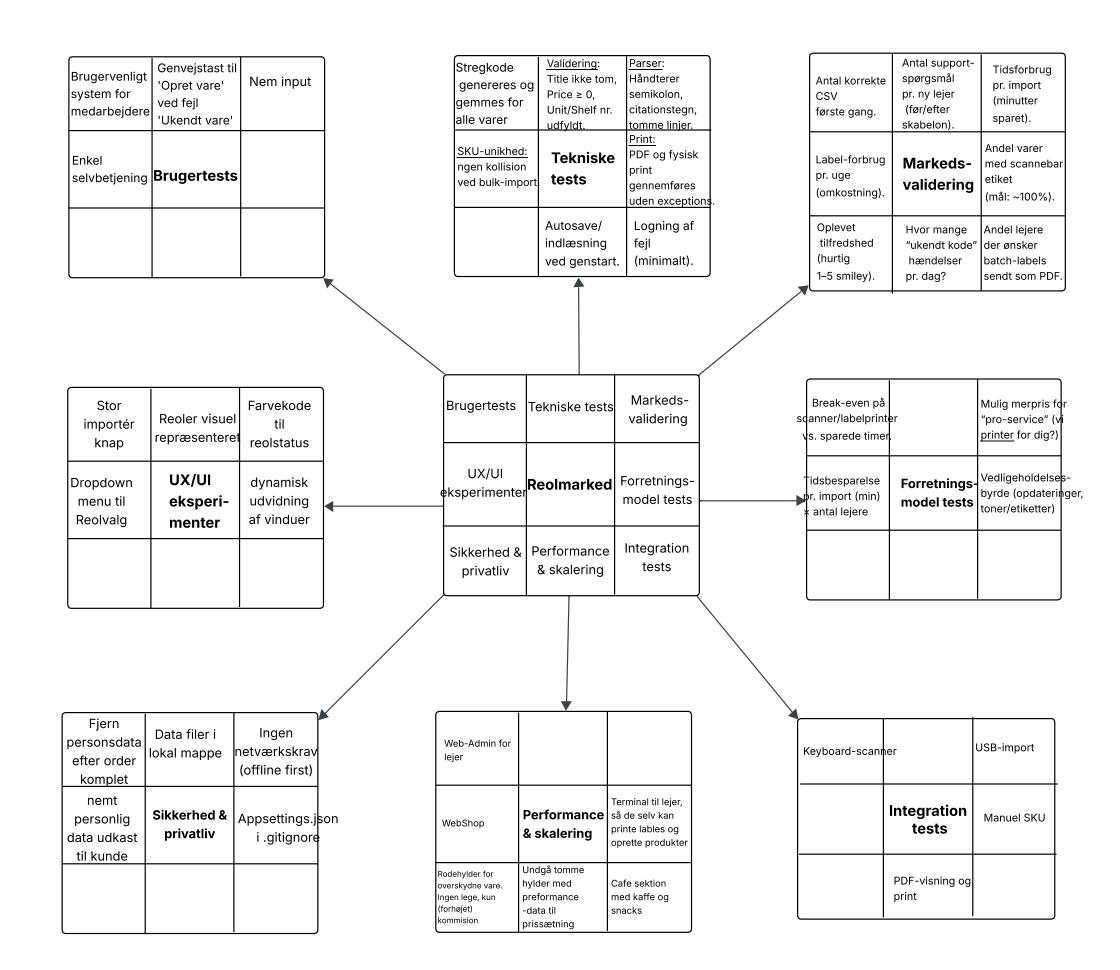


Double diamond



Første divergens:

- For meget administration
- Vil gerne tjene penge
- Vil gerne være midtpunkt i byen
- Bæredygtighed
- Skalerbarhed
- Stregkoder
- Systemet skal kunne bruges på en pc
- Rabatter



Decreasing churn to increase LTV	Increasing upfront prices	Targeting bigger clients
	Increasing revenue	Leveraging upselling
		Invoicing exceptional customer support

Optimizing resource utilization of servers	Migrating part of the data centers from AWS to OVH	Improving time complexity of software algorithms
	Increasing gross margin rate	Improving self- support platform

Stopping Facebook ads	Monitoring acquisition costs better	Improving development teams velocity	
	Reducing other var. costs	Reducing technical debts	

Refinancing high yield	Renegotiating Tranche A &	Increasing lease financing
Tranche	Tranche B rates	for new capex
	Decreasing debt service	Anticipating high yield Tranche reimbursement

Increasing revenue	Increasing gross margin rate	Reducing other variables cos
Decreasing debt service	Improving cash flows	Reducing fixe costs
Decreasing tax rate	Decreasing working capital	Increasing capex efficiency

Reducing surface lease rental thank to remote working	Stopping unnecessary travels	Centralizing IT procurement
	Reducing fixed costs	Renegotiating insurance premiums
	Restricing access to vehicle leasing	Changing phone operator

Apply for R&D tax exemptions	Creating a holding in Ireland	Redefining international transfer price policy
	Decreasing tax rate	

Developping downpayment	Renegotiating payment terms of main suppliers	
	Decreasing working capital	

Allocating capex better	Enriching capex decision- making	Organizing tenders with at least 3 potential suppliers
	Increasing capex efficiency	