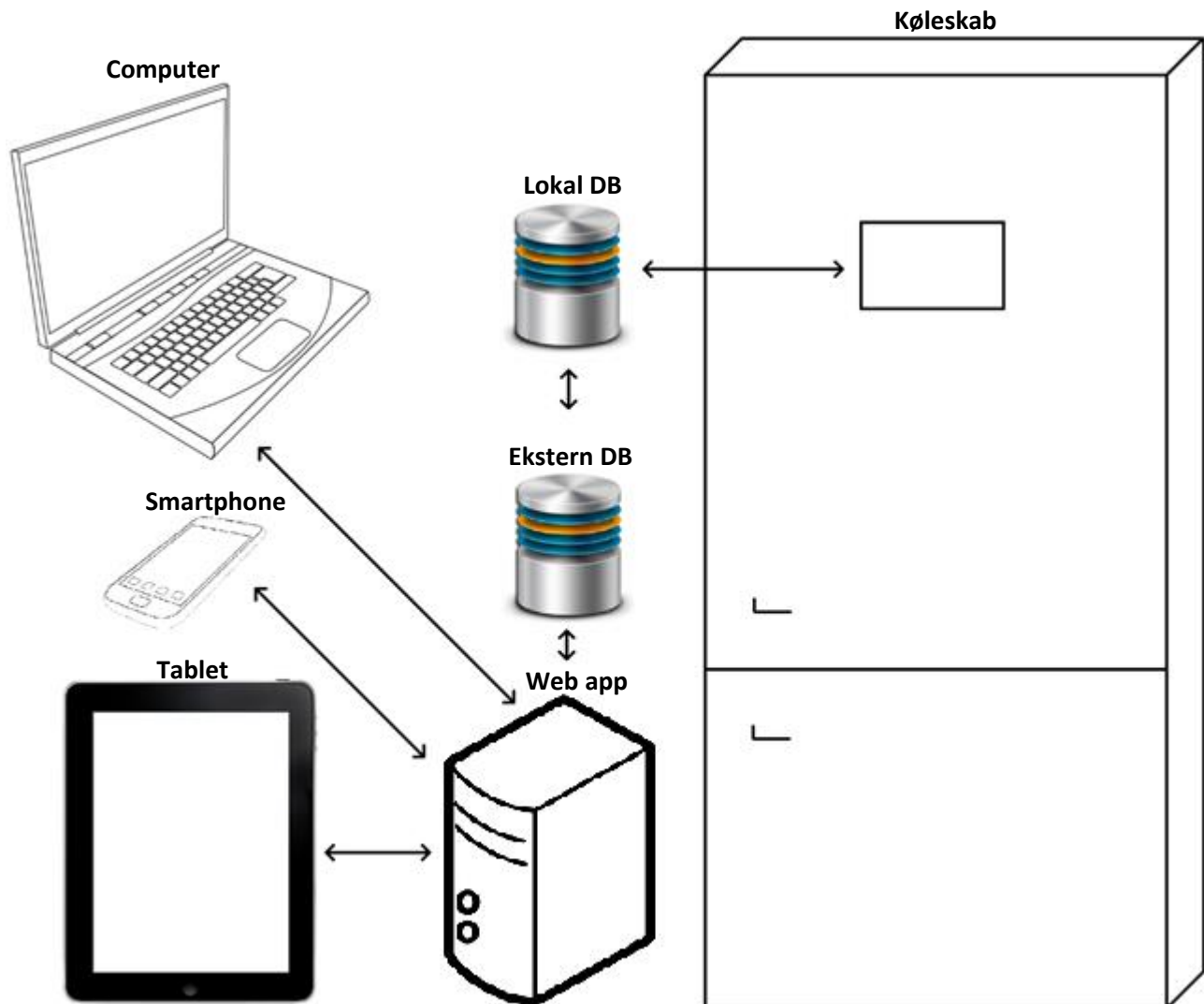


SmartFridge

Projektformulering



Figur 1 Skitsering af systemet SmartFridge

Formålet med dette projekt er at udvikle et system, som tillader registrering af varer i køleskab via en grafisk brugergrænseflade på en lokal skærm. Lagring af disse oplysninger sker i en lokal database, der synkroniseres med en ekstern database, som kan tilgås via et web-interface.

Systemet, som er relativt simpelt, byder desuden på rig mulighed for udvidelse, i form af inkorporering af ekstra funktioner. Eksempelvis oplysning om ernæringsværdier, opskrifter baseret på eksisterende ingredienser og eksisterende/fremtidige tilbud på manglende varer.

Systemet skal ses som en ekstern tilføjelse til eksisterende køleskabe, og ikke i første omgang som en indbygget feature i nye modeller. Dette sikrer at systemet kan tilbydes til en bredere målgruppe end køberne af "high end"-køleskabe.

Visionen er at tilbyde Bruger et hurtigt og effektivt overblik over køleskabets og fryserens indhold, til lettelse i en hverdag, hvor man ikke altid har en opdateret indkøbsliste inden for rækkevidde.

Systembeskrivelse

Systemet skal kunne assistere Bruger, ved tilbyde opretholdelse af en liste over hvilke varer, der er i køleskabet. Der kan tilføjes og fjernes varer i takt med indkøb og brug, samt opretholdes en liste over ønskede varer i køleskabet. Systemet vil desuden bestå af en lokal og en ekstern database, hvori de oprettede lister gemmes, så listerne også kan tilgås fra en web-applikation.

Når Bruger starter systemet, vil den lokale database synkronisere med den eksterne database, for at sikre at begge databaser er opdateret og stemmer overens. Herefter vil den lokale og den eksterne database synkronisere jævnligt, så længe systemet er i brug.

Når systemet er startet, vil det være muligt for Bruger at tilføje varer til listen over varer i køleskabet, på indkøbslisten eller i standardbeholdningen, hvorpå Bruger får eksisterende varetyper som forslag, eller kan tilføje nye varetyper selv.

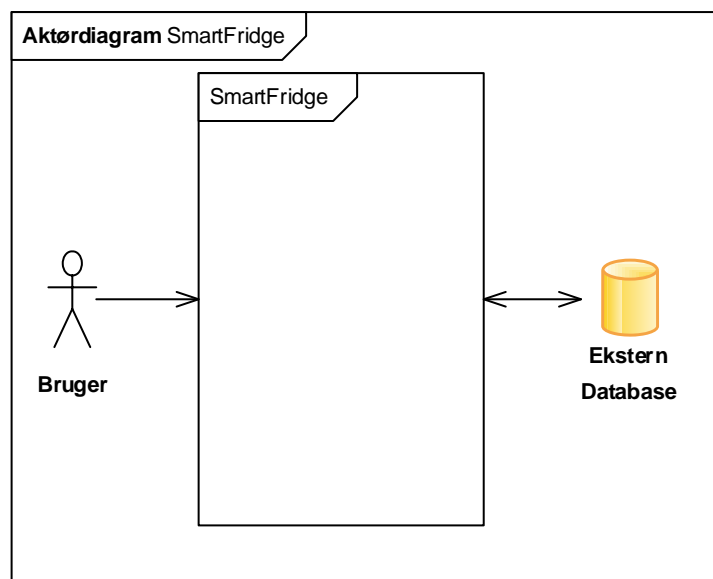
Det er muligt at se varerne på de forskellige lister. I denne forbindelse vil det være muligt at redigere mængden af varerne, i takt med at de fjernes.

Systemet vil påminde Bruger når køleskabet indeholder en mindre kvantitet af en vare, end der er angivet på listen over standard-varer, og det vil være muligt for Bruger at se hvad der mangler i køleskabet, samt en indkøbsliste, der som minimum indeholder de manglende varer.

Kravspecifikation

Aktører

Figur 2 viser use case-diagrammet med alle aktører og deres forhold til systemet SmartFridge.



Figur 2 Aktørdiagram over SmartFridge

Aktørbeskrivelse

Bruger

Aktørnavn:	Bruger
Alternativt navn:	
Type:	Primær
Beskrivelse:	Bruger er den primære bruger af systemet. Bruger interagerer med systemet gennem en brugergrænseflade.

Ekstern Database

Aktørnavn:	Ekstern Database
Alternativt navn:	Online Database
Type:	Sekundær
Beskrivelse:	Den Eksterne Database er en database der ligger på en webserver. Den har kun forbindelse til det lokale system når der synkroniseres (se Use Case 7), og fungerer som direkte database for web-app'en. Databasen får tilføjet, fjernet og redigeret data ved synkronisering, eller manipulationen kan foregå fra systemets web app.

Termliste

Fridge app

Fridge app er den lokale del af systemet, og dækker over den lokale brugergrænseflade, samt den lokale database.

Web app

Web app er den eksterne del af systemet, og dækker over websitet.

Kernefunktionalitet

Begrebet dækker over de mest basale funktionaliteter, som gør at systemet er sammenhængende og brugbart. Disse funktionaliteter er repræsenteret ved use cases 1-7.

Use cases

For alle use cases gælder det, at i enhver undermenu, findes der altid mulighed for at vende tilbage til hovedmenuen.

UC1: Opstart af applikation

Når Bruger tænder applikationen, startes programmet på enheden, og GUI-programmet starter i hovedmenuen. Programmets lokale database synkroniseres med den eksterne database, og indlæser alle gemte data.

Use case nr./navn	1 - Opstart af Applikation
Mål	At starte applikationen.
Initialisering	Bruger
Aktører	<ul style="list-style-type: none">- Bruger- Ekstern Database.
Samtidige forekomster	1
Forudsætninger	Applikationen er ikke startet.
Resultat	Applikationen er startet op, og hovedmenuen vises på skærmen.
Hovedscenarie	<ol style="list-style-type: none">1. Bruger tænder for systemet.2. Applikationens lokale database synkroniserer med den eksterne database. [Undtagelse 2.a: Ingen forbindelse til internettet] <ol style="list-style-type: none">3. Programmet starter i hovedmenuen.
Alternativt flow	
Undtagelser	2.a: Ingen forbindelse til internettet. 2.a.1 Applikationen fortsætter ved punkt 3.

UC2: Afslutning af applikation

Når Bruger afslutter applikationen, synkroniseres den lokale database med den eksterne, alle aktive processer afsluttes hensigtsmæssigt, og applikationen afsluttes.

Use case nr./navn	2 - Afslutning af applikation
Mål	At lukke ned for applikationen.
Initialisering	Bruger
Aktører	<ul style="list-style-type: none">- Bruger- Ekstern Database
Samtidige forekomster	1
Forudsætninger	UC1: Opstart af applikation
Resultat	Applikationen er lukket ned, og alle lokale ændringer er synkroniseret med den eksterne database.
Hovedscenarie	<ol style="list-style-type: none">1. Bruger trykker på afslut.2. Applikationens lokale database synkroniseres med den eksterne database. [Undtagelse 2.a: Ingen forbindelse til internettet] <ol style="list-style-type: none">3. Applikationens igangværende processer afsluttes.
Alternativt flow	
Undtagelser	2.a: Ingen forbindelse til internettet. 2.a.1 Applikationen fortsætter ved punkt 3.

UC3: Se varer

Bruger trykker på knappen "Se varer" på GUI, hvorefter Bruger præsenteres for de eksisterende lister; "I køleskab", "Standard-varer" og "Indkøbsliste". Bruger vælger "I køleskab", og præsenteres for en liste over hvilke varer, som i øjeblikket er registeret til at være i køleskabet. Til højre for den enkelte vare, fremgår mængden af varen, der befinder sig i øjeblikket. Sammen med antallet af varer på hver linje, findes to knapper, som hhv. kan øge og mindske antallet af varer.

Hvis Bruger, i stedet for "I køleskab", vælger en af de andre muligheder, vil samme type liste fremkomme, med de samme muligheder. På indkøbslisten vil de standard-varer, som ikke er i køleskabet i tilstrækkelig mængde, fremgå automatisk, og vil ikke kunne fjernes, men mængden kan øges. Desuden vil der være en knap, hvor det er muligt at nulstille listen, således at kun varer, som mangler i standardbeholdningen, vil restere.

Use case nr./navn	3 – Se varer
Mål	At få frembragt varerne på en liste.
Initialisering	Bruger
Aktører	- Bruger
Samtidige forekomster	1
Forudsætninger	UC1: Opstart af applikation
Resultat	Indholdet på den valgte liste vises på skærmen.
Hovedscenarie	<ol style="list-style-type: none"> 1. Bruger trykker på knappen "Se varer". 2. Bruger trykker på knappen "I køleskab". <p>[Alternativt flow 2.a: Bruger trykker på "Indkøbsliste"] [Alternativt flow 2.b: Bruger trykker på "Standard-varer"]</p> <ol style="list-style-type: none"> 3. En liste over nuværende varer i køleskabet, samt mængden af disse, vises på skærmen. <p>[Undtagelse 3.a: Listen over varer i køleskabet er tom]</p>
Alternativt flow	<p>2.a Bruger trykker på "Indkøbsliste".</p> <p>2.a.1 En liste over nuværende varer på indkøbslisten, samt mængden af disse, vises på skærmen.</p> <p>[Undtagelse 2.a.1.a: Indkøbslisten er tom]</p> <p>2.b Bruger trykker på "Standard-varer".</p> <p>2.b.1 En liste over nuværende standard-varer, samt mængden af disse, vises på skærmen.</p> <p>[Undtagelse 2.b.1.a: Listen over standard-varer er tom]</p>
Undtagelser	<p>3.a Listen over varer i køleskabet er tom.</p> <p>3.a.1 En besked om at listen over varer i køleskabet er tom, vises på skærmen, og UC afsluttes.</p> <p>2.a.1.a Indkøbslisten er tom.</p> <p>2.a.1.a.1 En besked om at indkøbslisten er tom, vises på skærmen, og UC afsluttes.</p> <p>2.b.1.a Listen over standard-varer er tom.</p> <p>2.b.1.a.1 En besked om at listen over standard-varer er tom, vises på skærmen, og UC afsluttes.</p>

UC4: Tilføj vare

Bruger trykker på knappen "Tilføj" på GUI. Der åbnes et nyt vindue, hvori forskellige valgmuligheder fremkommer, som tilsammen udgør en vare. Mange af felterne er valgfri.

Først anmodes Bruger om hvilken liste, varen skal indsættes på:

- Liste
 - Varebeholdning
 - Indkøbsliste
 - Standardbeholdning
 - Hvis varen allerede eksisterer på listen, lægges antallet blot sammen med det gamle antal.

Følgende informationer fremkommer:

- Varetype
 - Forslag til eksisterende varer fremkommer, mens Bruger skriver. Bruger vælger et af forslagene, eller lader det indtastede navn stå.
- Antal
 - Antal kan indtastes, eller der kan trykkes på to knapper som hhv. øger eller mindsker antallet af varen.
 - Som standard sat til 1 af standardenheden for den valgte vare, med mindre andet er angivet for varetypen (f.eks. 500g oksekød).
- Volumen/Vægt
 - Hvis der vælges en kendt varetype, vil programmet selv fylde nogle valgfri felter ud, fx Volumen/Vægt. Disse kan dog ændres.

Use case nr./navn	4 - Tilføj vare
Mål	At tilføje en vare til en liste.
Initialisering	Bruger
Aktører	<ul style="list-style-type: none">- Bruger- GUI
Samtidige forekomster	1
Forudsætninger	UC3: Se varer
Resultat	Varen er tilføjet til en liste.
Hovedscenarie	<ol style="list-style-type: none">1. Bruger trykker på "Tilføj".2. Varetype vælges. [Undtagelse 2.a: Varetypen eksisterer ikke] <ol style="list-style-type: none">3. Antal vælges.4. Volumen/vægt vælges.5. Bruger trykker på 'Tilføj og afslut,' og varen tilføjes til listen. [Alternativt flow 5.a: Bruger trykker på "Tilføj"] <ol style="list-style-type: none">6. Der returneres til listen.
Alternativt flow	5.a Bruger trykker på "Tilføj". 5.a.1 Varen gemmes, og der fortsættes fra punkt 2.
Undtagelser	2.a Varetypen eksisterer ikke. 2.a.1 Bruger indtaster den ønskede vare, og fortsætter fra punkt 3.

Kommende Use Cases

Kernefunktionalitet

Synkroniser til ekstern database

Applikationens lokale database synkroniseres med en ekstern database. Synkroniseringen initieres ved både systemstart og systemafslutning, og foretages desuden løbende under drift.

I applikationens hovedmenu findes en knap, som kan initiere en synkronisering mellem databaserne. Denne knap indikerer også hvorvidt der er foretaget ændringer på den lokale database, som endnu ikke er synkroniseret med den eksterne.

Rediger vare

Bruger trykker på knappen "Se varer" på GUI, hvorefter bruger præsenteres for de eksisterende lister; "I køleskab", "Standard-varer" og "Indkøbsliste". Bruger vælger den ønskede liste, og listens indhold vises. Bruger trykker på knappen rediger, og præsenteres for en ny menu, hvor der kan vælge mellem at ændre navnet på varen, og enheden som varen vises i (f.eks. kilo, gram, liter). Bruger vælger at ændre navnet, og præsenteres for en tekstboks, hvor det nuværende navn står. Bruger retter navnet og trykker "Udført", hvorefter ændringen gemmes, og forrige menu fremkommer.

Havde det været enheden, der skulle ændres, havde Bruger valgt enhed, hvorefter Bruger præsenteres for de mulige enheder. Den ønskede enhed vælges, og Bruger trykker "Udført" og bringes herefter tilbage til forrige menu.

Fjern vare

Bruger trykker på knappen "Se varer" på GUI, hvorefter bruger præsenteres for de eksisterende lister; "I køleskab", "Standard-varer" og "Indkøbsliste". Bruger vælger "I køleskab", og præsenteres for en liste over hvilke varer, der i øjeblikket er registreret til at være i køleskabet. Ved siden af varen, findes en knap, som Bruger trykker på, og varen forsvinder fra listen. Bruger trykker på knappen for alle varer som ønskes fjernet.

Hvis Bruger, i stedet for "I køleskab", vælger en af de to andre knapper, vil samme type liste fremkomme, med de samme muligheder. Fjernes en vare, som forekommer på standard-listen, fra indkøbslisten, vil forskellen mellem standard-listen og varebeholdningen stadig figurere. Ønskes varer fra standard-listen fjernet, fjernes disse fra listen "Standard-varer".

Udvidelser

Følgende udvidelser vil senere blive opdelt efter MoSCoW¹-metoden. De vil blive implementeret efter prioritet, efter systemets kernefunktionaliteter er implementeret.

Sidste type vare fjernes

Når Bruger fjerner den sidste type af en vare, kommer applikationen med en notifikation, for at gøre opmærksom på dette.

Gammel vare fjernes

Hvis holdbarhedsdatoen er overskredet ift. den dato som Bruger har angivet for en vare, kommer notificeres Bruger. Notifikationen fjernes når Bruger indikerer til systemet at varen er fjernet, eller holdbarhedsdatoen på varen er ændret til en fremtidig dato.

¹ http://en.wikipedia.org/wiki/MoSCoW_method

Tilføj opbevaringssted

Bruger vælger tilføj opbevaringssted, og giver denne et navn. Bruger kan nu vælge dette sted som opbevaringsplads, når en ny vare tilføjes eller redigeres.

Log in

Når Bruger vil tilgå web-app'en, skal der først logges ind. Når bruger er logget ind, kan egen del af den eksterne database tilgås, og de sædvanlige funktioner vil være tilgængelige.

Scan vare

En strekkodescanner tilsluttes systemet, og varer kan scannes. Varerne tilføjes til en valgfri varebeholdningsliste.

Vis ernæringsværdier

Bruger kan få oplyst ernæringsværdier for de enkelte varer.

Find opskrift

Bruger kan finde opskrifter baseret på de tilgængelige varer. Applikationen kan ud fra den valgte opskrift danne en indkøbsliste, med evt. manglende varer.

Valg af tema

Bruger får mulighed for at skifte grafisk tema på applikationen.

Juster temperaturalarm

Et termometer, som kan kommunikere med Fridge app'en, lægges i køleskabet, og Bruger sætter en max.- og en min.-temperatur. Kommer temperaturen uden for de satte værdier, advares Bruger.

Find tilbud

Bruger vælger "Find tilbud" på den færdige indkøbsliste. Applikationen finder nu selv de supermarkeder hvor der er tilbud på de varer, der er på indkøbslisten. De butikker hvor alle nødvendige varer sammenlagt kan købes billigst, kommer først.

Ikke-funktionelle krav

Alle krav er specificeret ud fra Lenovo Yoga 2 Pro² som platform.

1. System

- 1.1. Funktionaliteten for Fridge app og web app skal være ens.

2. Databaser

- 2.1. Den lokale og den eksterne database skal automatisk synkroniseres hvert 10. minut.
- 2.2. I tilfælde af konflikter ved synkronisering, overskriver de nyest tilføjede data de ældste.

3. Fridge app

- 3.1. Ændringer af data lagres straks i den lokale database.
- 3.2. En knap/et ikon på skærmen skal indikere status for synkronisering.
 - 3.2.1. Synkroniseret
 - 3.2.2. Ikke synkroniseret
 - 3.2.3. Ingen forbindelse
- 3.3. Responstiden for navigation må maksimalt være to sekunder.
- 3.4. Skal kunne anvendes uden internetforbindelse.

4. Web app

- 4.1. Ændringer af data lagres straks i den eksterne database.

² <http://www.wupti.com/file/file?fileId=266584> (tilføjes i endelig rapport som bilag)

Accepttest

Prækondition: Systemet er slukket

UC1: <i>Opstart af applikation</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 1: Bruger tænder for systemet.	Bruger starter app'en, og der testes visuelt om app'en starter op.	App'en starter op.		
Punkt 2: Applikationens lokale database synkroniserer med den eksterne database.	Bruger tilføjer en vare til en liste gennem web-app'en. App'en startes, og der testes visuelt, om den nye vare er tilføjet i den lokale database.	Den lokale og den eksterne database er blevet synkroniseret, og indholdet er ens.		
Punkt 3: Programmet starter i hovedmenuen.	Der inspiceres visuelt om hovedmenuen er kommet frem.	Hovedmenuen vises.		

UC1: <i>Undtagelser</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 2.a: Ingen forbindelse til internettet.	Indholdet på en liste på Fridge app'en kendes i forvejen. Fridge app'en slukkes. En vare tilføjes til en liste via web-app'en. Fridge-app'en startes op uden internetforbindelse. Det testes visuelt om indholdet er opdateret.	Listen er som før, og den nye vare er ikke tilføjet til listen på Fridge-app'en.		

Prækondition: Applikationen er startet

UC2: <i>Afslutning af applikation</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 1: Bruger trykker på afslut.	Bruger trykker på knappen beregnet til at afslutte app'en.	App'en lukkes ned.		
Punkt 2: Applikationens lokale database synkroniseres med den eksterne database.	Bruger tilføjer en vare til en liste, og afslutter app'en. Der testes visuelt gennem web-app'en, om den nye vare er tilføjet i den eksterne database.	Den lokale og eksterne database er blevet synkroniseret, og indholdet er ens.		

UC2: <i>Undtagelser</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 2.a: Ingen forbindelse til internettet.	Forbindelse til internettet fjernes. Bruger tilføjer en vare til en liste, og afslutter app'en. Der testes visuelt gennem web-app'en, om den nye vare er tilføjet i den eksterne database.	Den lokale og eksterne database er ikke blevet synkroniseret, og indholdet på den eksterne database er som før nedlukningen af app'en.		

Prækondition: Systemet er tændt (UC1), og mindst én vare er tilføjet til listen "I køleskabet" (UC4).

UC3: <i>Se varer</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 1: Bruger trykker på knappen "Se varer".	Bruger trykker på knappen "Se varer", og tester visuelt om rette menu fremkommer.	Et nyt vindue, med oversigt over de forskellige lister, vises på skærmen.		
Punkt 2-3: Bruger trykker på knappen "I køleskab".	Bruger trykker på knappen "I køleskab", og tester visuelt om den korrekte liste fremkommer.	En liste over nuværende varer i køleskabet, samt mængden af disse, vises på skærmen.		

Prækondition: Systemet er tændt (UC1), og mindst én vare er tilføjet til hver af listerne "Indkøbsliste" og "Standard-varer" (UC4).

UC3: <i>Alternativt flow</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 2.a: Bruger trykker på "Indkøbsliste".	Bruger trykker på knappen "Indkøbsliste", og tester visuelt om den korrekt liste fremkommer.	En liste over nuværende varer på indkøbslisten, samt mængden af disse, vises på skærmen.		
Punkt 2.b: Bruger trykker på "Standard-varer".	Bruger trykker på knappen "Standard-varer", og tester visuelt om den korrekt liste fremkommer.	En liste over nuværende standard-varer, samt mængden af disse, vises på skærmen.		

Prækondition: Systemet er tændt (UC1), og alle lister er tomme.

UC3: <i>Undtagelser</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 3.a: Listen over varer i køleskabet er tom.	Bruger trykker på knappen "I køleskab", og tester visuelt om den korrekt meddelelse fremkommer.	En besked om at listen over varer i køleskabet er tom, vises på skærmen.		
Punkt 2.a.1.a: Indkøbslisten er tom.	Bruger trykker på knappen "Indkøbsliste", og tester visuelt om den korrekt meddelelse fremkommer.	En besked om at indkøbslisten er tom, vises på skærmen.		
Punkt 2.b.1.a: Listen over standard-varer er tom.	Bruger trykker på knappen "Standard-varer", og tester visuelt om den korrekt meddelelse fremkommer.	En besked om at listen over standard-varer er tom, vises på skærmen.		

Prækondition: Systemet er tændt (UC1), og en liste over varer vises på skærmen (UC3).

UC4: <i>Tilføj Vare</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 1: Bruger trykker på "Tilføj".	Bruger trykker på knappen "Tilføj", og tester visuelt om det korrekte vindue fremkommer.	Et vindue med valgmuligheder vises på skærmen.		
Punkt 2: Varetype vælges.	Bruger vælger varetypen "Mælk".	Den valgte varetype markeres som valgt.		
Punkt 3: Antal vælges.	Bruger vælger mængden 1 stk.	Mængden 1 er valgt.		
Punkt 4: Volumen/vægt vælges	Bruger vælger 1 liter.	Volumenen "1 liter" er valgt.		
Punkt 5-6: Bruger trykker på "Tilføj og afslut", og varen tilføjes til listen.	Bruger trykker på "Tilføj og afslut", og tester visuelt om varen er tilføjet til listen.	Den tilføjede varer figurerer på den viste liste.		

UC4: <i>Alternativt flow</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 5.a: Bruger trykker på "Tilføj"	Bruger trykker på "Tilføj", og tester visuelt om det korrekte vindue fremkommer.	Et vindue med valgmuligheder vises på skærmen.		

UC4: <i>Undtagelser</i>	Test	Forventet resultat	Resultat	Godkendt/ kommentar
Punkt 2.a: Varetypen eksisterer ikke.	Bruger indtaster en varetype som ikke findes i systemet, og tester visuelt om varen blive stående.	Den nye varetype er valgt.		