# Projektformulering

Opgaven i 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 brugeren 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 brugeren ved lade dem opretholde en liste over hvilke varer de har i deres køleskab. Der kan tilføjes og fjernes varer i takt med indkøb og brug, samt opretholde en liste over ønskede varer i køleskabet. Systemet vil også bestå af en lokal og ekstern database, hvori de oprettede lister vil blive gemt, så listerne også kan tilgås fra en web applikation.

Når brugeren starter systemet, vil den lokale database synkroniserer med den eksterne database for at sikre at listerne er opdateret, hvis der er sket ændringer fra webapplikationen eller omvendt. Derefter vil den lokale database og eksterne synkronisere jævnligt, imens systemet er i brug.

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

Det er muligt at kunne se varerne på køleskabslisten og standardbeholdningen. Når man ser varerne, vil det være muligt at kunne redigere de varer man har tilføjet og at fjerne dem, når man f.eks. har brugt en liter mælk.

Systemet vil påminde brugeren om når der mangler noget i køleskabet, i forhold til standardbeholdningen, hvilket vil ske via mail og en påmindelse på app’en og de vil også være muligt for brugeren at kunne se hvad der mangler i køleskabet, samt brugerens indkøbsliste.

Når Bruger afslutter applikationen, synkroniseres den lokale database med den eksterne, og applikationen vil blive afsluttet.

# Kravspecifikation

## Aktører

På figuren til højre ses use case-diagrammet med alle aktører og deres forhold til systemet SmartFridge.



## Aktørbeskrivelse

### Bruger

|  |  |
| --- | --- |
| **Aktørnavn:** | Bruger |
| **Alternativt navn:** | - |
| **Type:** | Primær |
| **Beskrivelse:** | Bruger er den primære bruger af systemet. Bruger interagere med systemet vha. af en enhed der har applikationen installeret. Interaktionen initialisere Use Case 1 til 7. |

### 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 systemet når der synkroniseres mellem den og den lokale database (se Use Case 7). Databasen får tilføjet, fjernet og redigeret data ved synkroniseringen, men manipulationen kan også foregå fra en webapplikation. |

## Termliste

### Første term

Beskrivelse

### Anden term

Beskrivelse

## Use cases

Generelle noter – f.eks. ”For alle use cases gælder det, at i enhver undermenu, findes der altid mulighed for at annullere og vende tilbage til hovedmenuen.”

### Opstart af applikation

|  |  |
| --- | --- |
| **Use case nr./navn** | 1- Opstart af Applikation |
| **Mål** | Starte applikationen. |
| **Initialisering** | Bruger |
| **Aktører** | * Bruger, Ekstern Database. |
| **Samtidige forekomster** | 1 |
| **Forudsætninger** | Applikationen er ikke allerede startet. |
| **Resultat** | Applikationen er startet op. |
| **Hovedscenarie** | 1. Brugeren tænder for systemet. 2. Applikationens lokale database synkronisere med den eksterne database. 3. programmet starter i GUI hovedmenuen. |
| **Undtagelser** | 2a. Der er ingen forbindelse til internettet.   1. Applikationens lokale database kan ikke synkronisere med den eksterne database ved opstart. 2. Applikationen fortsætter til punkt 3. |
| **Alternativt flow** | 1. Brugeren tilgår den tilhørende webapplikation. 2. Brugeren indtaster id. 3. Brugeren præsenteres for en tilsvarende GUI hovedmenu som i punkt 3 i Hovedscenariet. |

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.

### Afslutning af applikation

|  |  |
| --- | --- |
| **Use case nr./navn** | 2 - Afslutning af Applikation |
| **Mål** | Lukke ned for applikationen. |
| **Initialisering** | Bruger |
| **Aktører** | * Bruger, Ekstern Database. |
| **Samtidige forekomster** | 1 |
| **Forudsætninger** | Applikationen er startet op. |
| **Resultat** | Applikationen er lukket ned. |
| **Hovedscenarie** | 1. Brugeren trykker på afslut. 2. Applikationens lokale database synkroniseres med den eksterne database. 3. Igangværende processer afsluttes på en fornuftig måde. |
| **Undtagelser** | 2a. Der er ingen forbindelse til internettet.   1. Applikationens lokale database kan ikke synkronisere med den eksterne ved opstart. 2. Applikationen fortsætter til punkt 3. |
| **Alternativt flow** | 1. Brugeren har startet den tilhørende webapplikation. 2. Brugeren trykker afslut. 3. Brugeren bringes tilbage til webapplikationens startside, hvor id kan indtastes. |

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

### Tilføj Vare

|  |  |
| --- | --- |
| **Use case nr./navn** | 3 - Tilføj Vare |
| **Mål** | At tilføje en vare til en beholdning |
| **Initialisering** | Bruger |
| **Aktører** | * Bruger, GUI |
| **Samtidige forekomster** | 1 |
| **Forudsætninger** | UC1 |
| **Resultat** | Varen er tilføjet til en beholdning |
| **Hovedscenarie** | 1. Bruger trykker på ”Tilføj” på GUI. 2. Bruger vælger hvilken beholdning den fremtidige vare skal indsættes på og trykker ’OK’. 3. Varetype vælges.    1. Varetypen indtastes 4. Antal vælges. 5. Volumen/Vægt vælges. 6. Bruger trykker på ’Tilføj’ og varen tilføjes til den tidligere valgte beholdning 7. Bruger returneres til vinduet i punkt 3. |
| **Undtagelser** | 3a: Varetypen indtastes, der fortsættes fra punkt 4. |
| **Alternativt flow** |  |

Bruger trykker på knappen ”Tilføj” på GUI. Der åbnes et nyt vindue, hvori der kommer forskellige valgmuligheder, som 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 i den beholdning 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.
* Volumen/Vægt
  + Hvis der vælges en kendt varetype, vil programmet selv fylde nogle valgfri felter ud, fx Volumen/Vægt.

Som standard sat til 1 af standardenheden for den valgte vare, med mindre andet er angivet for varetypen (f.eks. 500g oksekød).

### Se varer

|  |  |
| --- | --- |
| **Use case nr./navn** | 5 – Se varer |
| **Mål** | At få frembragt en liste over varerne på en liste. |
| **Initialisering** | Bruger |
| **Aktører** | * Bruger |
| **Samtidige forekomster** | 1 |
| **Forudsætninger** | UC1 |
| **Resultat** | En liste over den nuværende varebeholdning vises på skærmen. |
| **Hovedscenarie** | 1. Bruger trykker på knappen ”Se varer”. 2. Bruger trykker på knappen ”I køleskab”.   **[Alternativt flow 2.a: Bruger trykker på ”Indkøbsliste”.]**  **[Alternativt flow 2.b: Bruger trykker på ”Standard-varer”.]**   1. En liste over nuværende varer i køleskabet, samt mængden af disse, vises på skærmen, og UC5 afsluttes.   **[Undtagelse 3.a: Listen over varer i køleskabet er tom.]** |
| **Alternativt flow** | **2.a Bruger trykker på ”Indkøbsliste”.**  2.a.1 En liste over nuværende varer på indkøbslisten, samt mængden af disse, vises på skærmen, og UC5 afsluttes.  **[Undtagelse 2.a.1.a: Indkøbslisten er tom.]**  **2.b Bruger trykker på ”Standard-varer”.**  2.b.1 En liste over nuværende standard-varer, samt mængden af disse, vises på skærmen, og UC5 afsluttes.  **[Undtagelse 2.b.1.a: Listen over standard-varer er tom.]** |
| **Undtagelser** | **3.a Listen over varer i køleskabet er tom.**  3.a.1 En besked om at listen over varer i køleskabet er tom, vises på skærmen, og UC5 afsluttes.  **2.a.1.a Indkøbslisten er tom.**  2.a.1.a.1 En besked om at indkøbslisten er tom, vises på skærmen, og UC5 afsluttes.  **2.b.1.a Listen over standard-varer er tom.**  2.b.1.a.1 En besked om at listen over standard-varer er tom, vises på skærmen, og UC5 afsluttes. |

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.

### Kommende Use Cases

### Kerne funktionalitet

## 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 vedkommende bringes tilbage til forrige menu. 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 derefter tilbage til forrige menu.

## User story: 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, som i øjeblikket er registeret til at være i køleskabet. Ved siden af varen, findes en knap, som Bruger trykker på, og varen forsvinder fra listen. Brugeren 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 fra indkøbslisten, som forekommer på standard-listen, vil forskellen mellem standard-listen og varebeholdningen stadig figurere. Ønskes varer fra standard-listen fjernet, fjernes disse fra listen ”Standard-varer”.

### Udvidelser

1. Mangel-af-vare-påmindelse

**UC 8 – Sidste type vare fjernes**

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

1. Udløbsdato/holdbarhedsdato notifikation

**UC 9 – Gammel vare fjernes**

Datoen er overskrevet den dato bruger har skrevet ind som sidste holdbarhedsdato for en vare. Applikationen kommer med en notifikation til brugeren. Notifikationen fjernes når brugeren indikerer til systemet at varen er fjernet.

1. Tilføjelse af flere skabe/frysere

**UC 10 – Tilføj opbevaringssted**

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

1. Webinterface (og evt. lokalt) log in

**UC 11 – Log in**

Når bruger vil tilgå webinterfacet skal der først logges ind. Når bruger er logget ind kan han tilgå sin egen database, og se hvad han har i sit eget køleskab.

1. Scanner

**UC 12 – Scan vare**

Bruger tager applikationens integrerede stregkode scanner og scanner sin vare. Denne er nu tilføjet til den lokation bruger har valgt

1. Ernæringsværdier

**UC 13 – Se vare V2**

Brugeren kan for hver vare også se hvilken ernæring der er i de enkelte vare. Eks.: Protein, fedt og kulhydrater, samt mængden af kalorier.

1. Opskrifter

**UC 14 – Find opskrift**

Bruger vælger find opskrift, og kan dernæst vælge om der skal findes opskrifter ud fra de vare bruger har indtastet i programmet, eller om bruger selv vil finde en opskrift. Applikationen kan ud fra den valgte opskrift danne en indkøbsliste, hvor der ses bort fra de varer bruger allerede har.

1. GUI: Køleskab ses som inventory med drag-n-drop items.

**UC 15 – Valg af tema**

Bruger får muligheden for at skifte tema på applikation, og kan vælge et mere interaktivt tema.

Eksempel tema:

* Bruger ser et køleskab på applikationen, og kan herefter vælge de enkelte vare fra en liste og drag-n-drop disse varer for at tilføje dem til lager beholdningen.

1. Temperatur bluetooth

**UC 16 – Justér temperatur alarm**

Bruger får koblet et blootooth termometer til sin applikation, og sætter en max og min temperatur. Kommer temperaturen over max eller under min. Kommer applikationen med en notifikation

1. Tilbud af ting på indkøbsliste/mangelliste

**UC – Find tilbud**

Brugeren 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 der sammenlagt kan købes billigst kommer først.

## Ikke funktionelle krav

Alle krav er specificeret ud fra Lenovo Yoga 2 Pro[[1]](#footnote-1) som platform.

1. **System**
   1. Funktionaliteten for Fridge GUI og Web GUI skal være ens.
2. **Databaser**
   1. Den lokale og den eksterne database skal automatisk synkroniseres hvert 10. minut.
   2. I tilfælde af konflikter ved synkronisering, overskriver de nyest tilføjede data de ældste.
3. **Fridge app**
   1. Ændringer af data lagres straks i den lokale database.
   2. En knap/et ikon på GUI’en skal indikere status for synkronisering
      1. Synkroniseret
      2. Ikke synkroniseret
      3. Ingen forbindelse
   3. Responstiden for navigation må maksimalt være to sekunder.
   4. Skal kunne anvendes uden internetforbindelse.

1. **Web app**
   1. Ændringer af data lagres straks i den eksterne database.

# Accepttest

Prækondition: Systemet er slukket

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC1: *Opstart af Applikation*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 1:**  Brugeren Tænder for systemet. | Visuel test. Brugeren | Applikationen starter op. |  |  |
| **Punkt 2:**  Applikationens lokale database synkroniseres med den eksterne database. | Visuel test. Indholdet af den lokale og eksterne database inspiceres. | Den lokale og eksterne database er blevet synkroniseret, og indholdet er ens. |  |  |
| **Punkt 3:**  Programmet starter i GUI hovedmenuen. | Visuel test. Der inspiceres visuelt om GUI hovedmenuen er kommet frem. | GUI hovedmenuen vises. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC1: *Undtagelser*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 2.a:**  Der er ingen forbindelse til internettet og applikationen kan ikke synkronisere den lokale database med den eksterne. | Forbindelse til internettet fjernes før opstart. | Den lokale og den eksterne database blev ikke synkroniseret. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC1: *Alternativt flow*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 1:**  Brugeren tilgår den tilhørende webapplikation. | Visuel test. webapplikationens url tilgås via en browser, og det ses at den starter. | Det ses at webapplikationen starter op, og er klart til at modtage et id. |  |  |
| **Punkt 2:**  Brugeren indtaster id. | En testperson indtaster id | Id er indtastet. |  |  |
| **Punkt 3:**  Brugeren præsenteres for en tilsvarende GUI hovedmenu som i punkt 3 i hovedscenariet. | Visuel test. Webapplikationens hovedmenu sammenlignes med køleskabets hovedmenu. | De to hovedmenuer er ens. |  |  |

Prækondition: Applikationen er startet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC2: *Afslutning af Applikation*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 1:**  Brugeren trykker på afslut. | (Hvad foretages der, og hvordan testes det? Visuelt/auditivt/ whatever) | (Hvad bør testen resultere i?) |  |  |
| **Punkt 2:**  Applikationens lokale database synkroniseres med den eksterne database. |  | Den lokale og eksterne database er blevet synkroniseret, og indholdet er ens. |  |  |
| **Punkt 3:**  Igangværende processer afsluttes på en fornuftig måde. |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC2: *Undtagelser*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 2.a:**  Der er ingen forbindelse til internettet og applikationen kan ikke synkronisere den lokale database med den eksterne. | Forbindelse til internettet fjernes før opstart | Den lokale og den eksterne database blev ikke synkroniseret |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC2: *Alternativt flow*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 1:**  Brugeren har startet den tilhørende webapplikation. | Visuel test. Det ses at web applikationen er åben. | Webapplikationen er åben. |  |  |
| **Punkt 2:**  Brugeren trykker afslut. | En testperson trykker afslut. | Webapplikationen afslutter. |  |  |
| **Punkt 3:**  Brugeren bringes tilbage til webapplikationens startside, hvor id kan indtastes. | Visuel test. Det ses at webapplikationen har returneret til start siden, hvor id kan indtastes. | Webapplikationen har returneret til startsiden, hvor id kan indtastes |  |  |

Prækondition: UC1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC3: *Tilføj Vare*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 1:**  Bruger trykker på ”Tilføj” på GUI. | Der trykkes på knappen ’Tilføj’ på GUI’en. | Et nyt vindue åbnes |  |  |
| **Punkt 2:**  Bruger vælger hvilken beholdning den fremtidige vare skal indsættes på. | En beholdning fra en dropdown-menu vælges og ’OK’ trykkes på GUI’en. | Et nyt vindue åbnes. |  |  |
| **Punkt 3:**  Varetype vælges | En varetype fra en dropdown-menu vælges. |  |  |  |
| **Punkt 4:**  Antal vælges | Et antal mellem 1 og 99 vælges. |  |  |  |
| **Punkt 5:**  Volumen/Vægt vælges | En volumen eller vægt vælges hvor det indtastede tal er numerisk. |  |  |  |
| **Punkt 6:**  Bruger trykker på ’Tilføj’ og varen tilføjes til den tidligere valgte beholdning | Der trykkes på ’Tilføj’ på GUI’en. | Varen er tilføjet til den valgte beholdning. Se UC5. |  |  |
| **Punkt 7:**  Bruger returneres til vinduet i punkt 3. |  | Vinduet fra punkt 3 vises. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC3: *Undtagelser*** | **Test** | **Forventet resultat** | **Resultat** | **Godkendt/ kommentar** |
| **Punkt 3.a:**  Varetypen indtastes | Indtast et navn manuelt | Varetypen registreres og kan vælges fra dropdown-menuen. |  |  |

Prækondition: Systemet er tændt (UC1), og mindst én vare er tilføjet til listen ”I køleskabet” (UC3).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC5: *Se varer*** | **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:**  Bruger trykker på knappen ”I køleskab”. | Bruger trykker på knappen ”I køleskab”, og tester visuelt om den korrekt 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” (UC3).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC5: *Alternativt flow*** | **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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC5: *Undtagelser*** | **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. |  |  |

1. http://www.wupti.com/produkter/computer-og-it/baerbar-computer/baerbar-pc/lenovo-yoga-2-pro-13-i5-orange-n [↑](#footnote-ref-1)