

MealMate



Makro-tracker App

Kravspekifikation

Andreas | Bjarke | Daniel

Vejleder: Flemming Sørensen

Deadline: 19-03-2025

Lokation: TEC Ballerup

Indholdsfortegnelse

| | |
|----------------------------------|----------|
| Case-beskrivelse | 3 |
| Problemformulering | 4 |
| Kravspecifikation | 5 |
| Indledning | 5 |
| Formål | 5 |
| Læsevejledning | 5 |
| Generelle krav | 6 |
| Brugerregistrering | 6 |
| Dashboard | 6 |
| Makro tracking | 6 |
| Personlige mål | 6 |
| Eksterne grænseflade-krav | 7 |
| Bruger grænseflade | 7 |
| Hardware grænseflade | 7 |
| Software grænseflade | 7 |
| Krav til programmets ydelse | 7 |
| Kvalitetsfaktorer | 8 |
| Pålidelighed | 8 |
| Vedligeholdelsesvenlighed | 8 |
| Genbrugelighed | 8 |
| Brugervenlighed | 8 |
| Effektivitet | 8 |
| Aktør-kontekstdiagram | 9 |
| Aktørbeskrivelser | 9 |
| Admin | 9 |
| Bruger | 10 |
| API | 10 |
| Use Cases | 11 |
| Use Case beskrivelser | 13 |
| A.1 - Opret bruger | 13 |
| A.2 - Log ind | 14 |
| A.3 - Rediger bruger | 15 |
| A.4 - Slet bruger | 16 |
| A.5 - Authorization | 17 |
| B.1 - Scan næringsindhold | 18 |
| B.2 - Indtast fødevare manuelt | 19 |
| B.3 - Tilføj fødevare | 20 |
| B.4 - Godkend fødevare | 21 |
| B.5 - Rediger fødevare | 22 |
| B.6 - Slet fødevare | 23 |
| B.7 - Tilføj fødevare i database | 24 |

| | |
|----------------------------|----|
| C.1. - Anmod tracking data | 25 |
| C.2. - Hent tracking data | 26 |
| E/R Diagram | 27 |
| Mockup | 28 |
| Tidsplan | 29 |

Case-beskrivelse

Visionen med dette projekt er at udvikle en makro-tracker app, som kan hjælpe brugere med at overvåge deres daglige makro- og kostindtag. I dag hvor sundhed og fitness er i fokus, er der en stigende efterspørgsel efter værktøjer, der kan hjælpe individer med at tage kontrol over deres kost og ernæring. En makro-tracker app, kan være en god ressource for alle, der ønsker at leve et sundere liv.

Denne app skal kunne scanne næringsindholdet på et produkt, for at læse information om kalorier, fedtstoffer, proteiner og kulhydrater. Dette vil gøre det nemt for brugerne at få præcis information om deres madvarer uden at skulle indtaste data manuelt. Ved at automatisere denne proces kan brugerne spare tid og minimere fejl, hvilket øger appens anvendelighed og nøjagtighed.

Derudover skal appen kunne gemme og spore brugernes daglige indtag for at give indsigt i deres kostvaner og fremskridt over tid. Ved at have en omfattende oversigt over deres indtag over makronæringsstoffer, kan brugerne træffe mere informerede beslutninger om deres kost og justere deres madvaner for at opnå deres sundheds- og fitness mål.

Appen vil være et effektivt værktøj for alle, lige fra dem, der blot ønsker at spise sundere, til dem, der har specifikke mål som vægttab, muskelopbygning eller vedligeholdelse af en sund vægt. Med funktioner som kost-scanning af madvarer, registrering af daglig indtag, målsætning og sporing, vil denne app tilbyde en holistisk løsning til personlig kost- og ernæringsstyring.

Problemformulering

Der er mangel på en nem og effektiv måde at holde styr på sine makroer. Det kan gøre det udfordrende for individer at opnå og opretholde deres sundheds- og fitness mål. Traditionelle metoder til makro-tracking er ofte tidskrævende og komplekse, hvilket kan afskrække og demotivere brugere fra at fortsætte med at overvåge deres kostvaner.

Mange apps der kan gøre dette findes allerede, men mangler direkte funktionalitet, som kan tilpasses til individuelle brugeres behov og livsstil, eller har funktionerne bag en paywall.

- Hvordan kan vi udvikle og designe en brugervenlig samt intuitiv makro tracker app, til at gøre det nemt og effektivt for brugere at tracke deres daglige kostindtag?
- Hvordan kan vi gøre selve indtastningen og tracking af kostindtag simplere?
- Hvilke funktioner kan udvikles for at forskellige diæter kan på nemmest muligt vis trackes i appen?
- Hvordan kan vi tilføje funktioner som gør det lettere for brugeren at være motiveret til at bruge app'en på længere sigt?

Kravspecifikation

Indledning

Formål

Denne kravspecifikation omhandler udviklingen af en makro-tracker app, med det formål at hjælpe brugere med nemt og effektivt at holde styr på deres daglige kostindtag for at opnå og opretholde deres sundheds- og fitness mål. App'en skal kunne tilpasse individuelles behov og livsstil uden at skjule essentielle funktioner bag en paywall. Udover det skal app'en hjælpe med at motivere brugeren til at opnå sine mål.

Læsevejledning

Kravspecifikationen er opdelt i flere sektioner, der beskriver formål, systembeskrivelse, funktionalitet, begrænsninger, fremtidige udvidelsesmuligheder, brugerprofiler, krav til udviklingsforløbet og specifikke tekniske krav.

Generelle krav

Brugerregistrering

Brugere skal kunne oprette en profil med brugeroplysninger: email, navn, fødselsdato, vægt og mål - herunder vægtforøgelse, vægttab, vedligeholdelse.

Dashboard

Dashboardet skal indeholde en oversigt, der viser samlet kalorieindtag, resterende kalorier i forhold til dagligt mål og makro-næringsstof fordeling (kalorier, protein, kulhydrater, fedt).

Derudover skal den også indeholde grafiske fremstillinger af kalorieindtag over tid, såsom daglige, ugentlige og månedlige grafer.

Makro tracking

Mulighed for manuel indtastning af fødevarer med angivelse af vægt og makro indhold.

Integration med en fødevaredatabase, så brugere nemt kan søge og tilføje fødevarer.

Scanning af strekkoder på fødevarer for hurtig og præcis tilføjelse af kalorieindhold og næringsoplysninger.

Når en bruger tilføjer indtaget fødevarer fra database, skal det automatisk tracks i forhold til brugerens personlige daglige mål, som så skal kunne ses på dashboardet

Personlige mål

Brugere kan sætte og justere deres daglige mål baseret på deres personlige sundheds- og fitness mål.

Tilpasning af makronæringsstof mål (kalorier, protein, kulhydrater, fedt) baseret på brugerens præferencer og diæt.

Eksterne grænseflade-krav

Bruger grænseflade

Brugerne skal kunne benytte appen på en android smartphone/tablet eller iPhone/iPad.

Hardware grænseflade

En server der kan håndtere at hoste en API server udviklet i Node.js samt en MongoDB database.

Software grænseflade

Appen skal køres både på Android og iOS.

Krav til programmets ydelse

- Tracking mulighed – brugeren skal kunne se tilbage på sine daglige samlet kalorie indtag m.m.
- Brugeren skal kunne indtaste kostindtag manuelt eller bruge kameraet til at scanne næringsindhold på fysiske produkter
- App'en skal have en nem og intuitiv UI
- Bruger data skal gemmes i database, så man kan logge ind på andre enheder og se sine fremskridt
- API serveren skal kun kunne tilgås af brugerne til hente enkeltvarer og deres egne data og retter
- Anvendelse af JWT (JSON Web tokens) for sikker autentifikation og autorisation.
- Anvendelse af BCrypt library for at kryptere passwords

Kvalitetsfaktorer

Pålidelighed

Der skal være styr på at serveren ikke har downtimes, da det er vigtigt at brugerne kan tilgå og tilføje deres data når det passer dem.

Vedligeholdelsesvenlighed

App'en samt API skal bygges på sådan en måde at udvidelse af projektet kan forekomme uden unødvendige ændringer af den kode der allerede virker.

Genbrugelighed

For at det førnævnte kan opnås, skal koden i app'en være så modulbaseret som muligt og API'en skal være godt struktureret.

Brugervenlighed

Alt i app'en skal være let tilgængeligt og forståeligt for brugeren. Det vil gøres ved brug af ikoner og beskrivende tekst, så brugeren nemt kan forstå hvad de navigerer til eller udfører.

Effektivitet

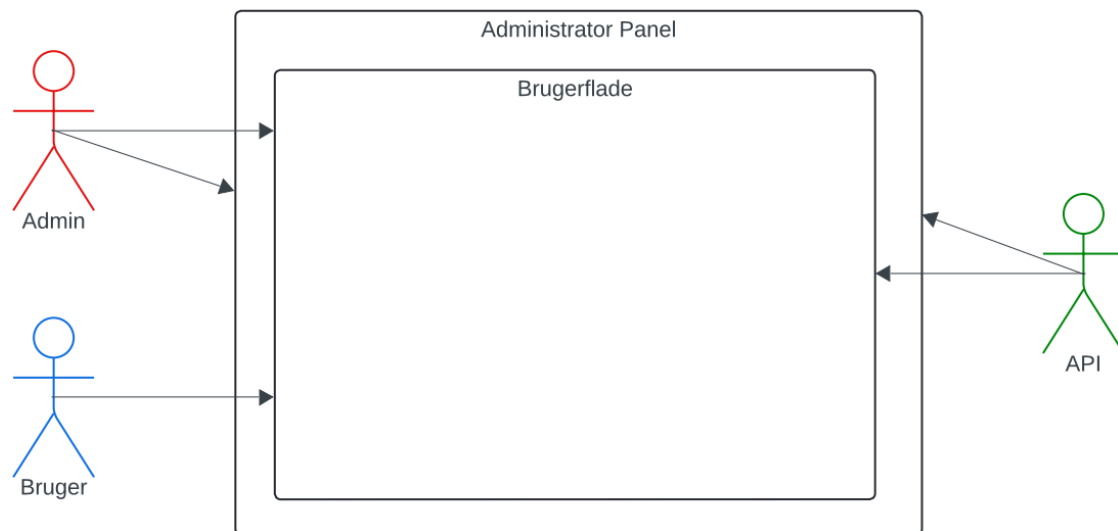
For at sørge for effektivitet, er API'en en REST-API, så flere brugere kan udføre requests på samme tid uden for lang ventetid.

Aktør-kontekstdiagram

Diagrammet viser hvordan de forskellige aktører interagerer med selve applikationen.

Det inkluderer et black box system, og 2 primære aktører & en sekundær aktør.

Aktørbeskrivelser er forklaret i følgende aktørbeskrivelser under diagrammet



Aktørbeskrivelser

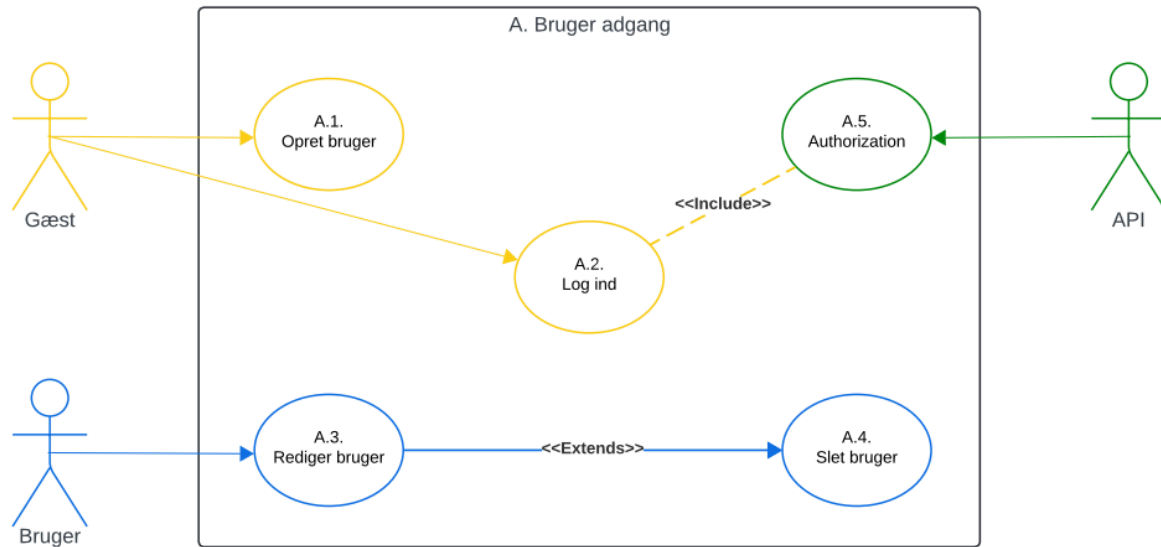
| | |
|---|---|
| Aktør navn | Admin |
| Type (person/maskine, primær/sekundær) | Er en person aktør, der er primær aktør i kalorie tracker administration. |
| Beskrivelse | Personen kan oprette, redigere og slette produkter. Når der bliver registreret et nyt produkt, kan det godkendes af administratoren, hvis det skal vises på listen til alle. |
| Antal samtidige aktører | 1-3 |

| | |
|---|--|
| Aktør navn | Bruger |
| Type (person/maskine, primær/sekundær) | Er en person aktør, der er primær aktør i kalorie tracker. |
| Beskrivelse | Brugere kan lave sin egen kalorieplan, se næringsindhold på produkter. |
| Antal samtidige aktører | 10-1000 |

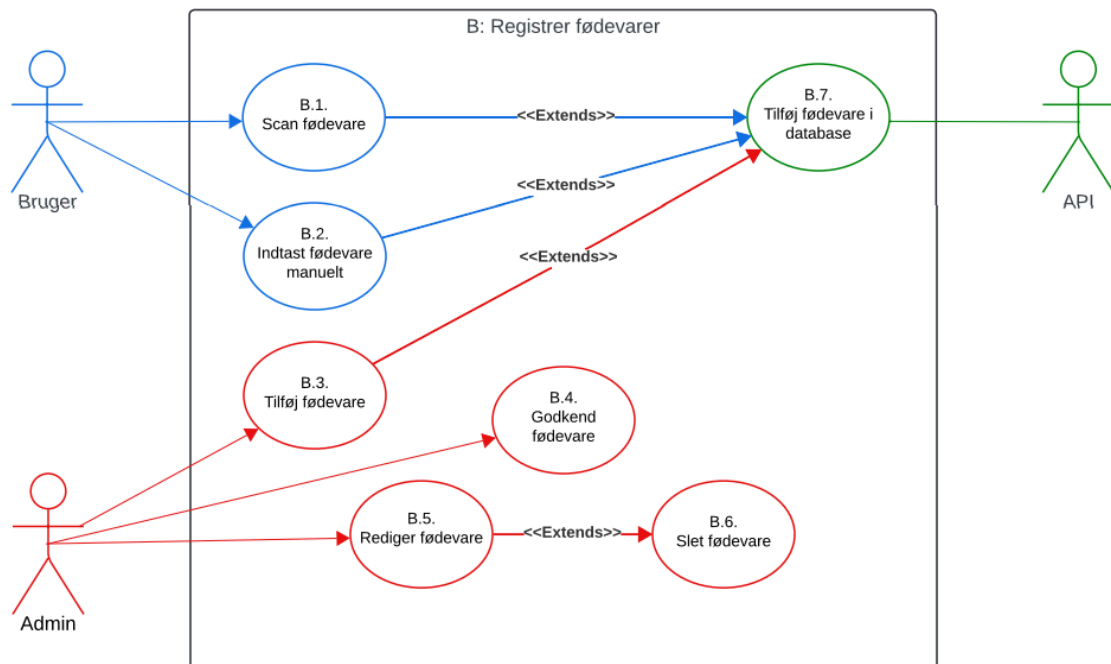
| | |
|---|---|
| Aktør navn | API |
| Type (person/maskine, primær/sekundær) | Er en maskine aktør, der er primær aktør i kalorie tracker. |
| Beskrivelse | API bruges til at sende data om produkter, bruger data, samt bekræfte authentication til login. Den bruges også til at gemme produkter, bruger data. |
| Antal samtidige aktører | 1 |

Use Cases

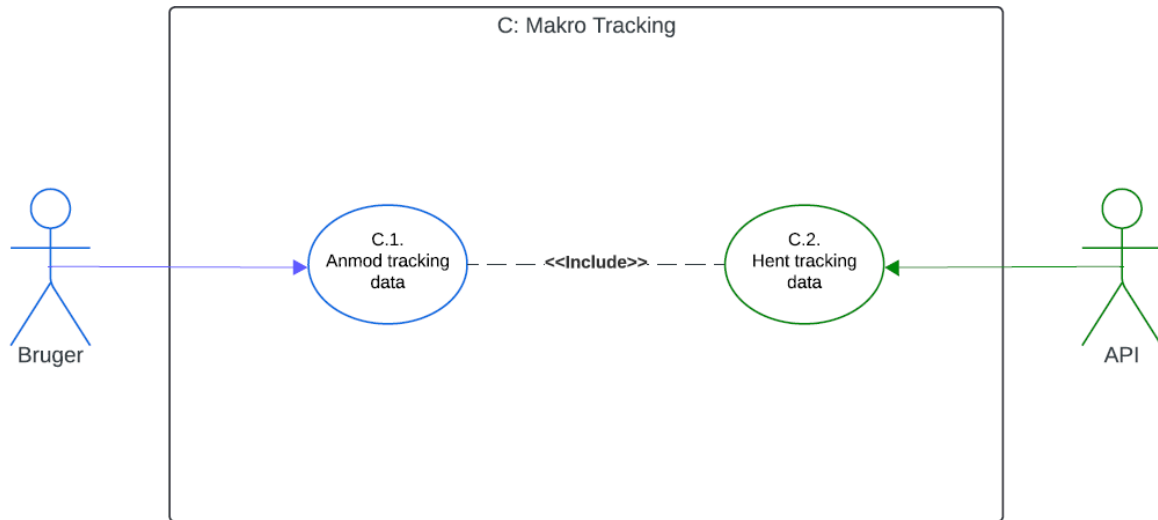
System A: Bruger adgang:



System B: Registrer fødevare:



System C: Makro Tracking



Use Case beskrivelser

| | | | |
|--------------------|---|---|----------|
| Use Case ID: | A.1 - Opret bruger | | |
| Mål | Bruger bliver succesfuldt oprettet | | |
| Initiering | Initieres af gæst | | |
| Aktør(er) | Gæst | | |
| Type | Selvstændig: X | Include: | Extends: |
| Frekvens: | 1 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | Bruger logges ind med det samme | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har åbnet appen og trykket på "Opret bruger" Aktøren har ikke oprettet en bruger i forvejen med deres e-mail | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktøren indtaster deres e-mail 2. Aktøren indtaster det ønskede password 3. Aktøren gentager det ønskede password 4. Aktøren indtaster sit ønskede brugernavn 5. Aktøren klikker "Bekræft" | | |
| Variationer | - | | |
| Slutresultat | Succes | Aktøren får oprettet en bruger og bliver logget ind | |
| | Fejl | Aktøren har indtastet en ugyldig e-mail Aktøren har indtastet en e-mail der i forvejen bliver brugt til en bruger Aktøren har indtastet et ugyldigt password Aktøren har gentaget password forkert | |

| | | | |
|--------------------|--|--|----------|
| Use Case ID: | A.2 - Log ind | | |
| Mål | Gæst bliver succesfuldt logget ind som bruger | | |
| Initiering | Initieres af gæst | | |
| Aktør(er) | Gæst | | |
| Type | Selvstændig: X | Include: A.5 | Extends: |
| Frekvens | 10 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har en bruger i forvejen API og database serverne er oppe og køre | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktør indtaster e-mail 2. Aktør indtaster password 3. Aktør klikker "log ind" 4. Log ind request sendes til API Use case "A.4 - Authorization" initieres 5. Aktør modtager svar fra API | | |
| Variationer | Variation 1 Aktør har ikke bruger i forvejen <ol style="list-style-type: none"> 1. Aktør klikker "Opret bruger" Use case "A.1 - Opret bruger" initieres | | |
| Slutresultat | Succes | Aktør bliver succesfuldt logget ind ved godkendelse fra API | |
| | Fejl | Aktør bliver ikke logget ind - modtager: <ul style="list-style-type: none"> - Aktør har skrevet forkert e-mail eller password - Authorization fejler | |

| | | | |
|--------------------|---|---|----------|
| Use Case ID: | A.3 - Rediger bruger | | |
| Mål | Bruger bliver succesfuldt redigeret | | |
| Initiering | Initieres af bruger | | |
| Aktør(er) | Bruger | | |
| Type | Selvstændig: X | Include: | Extends: |
| Frekvens | 10 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | Opdateringer skal kunne ses direkte efter gennemførelse | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har en bruger i forvejen | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktøren kan vælge at redigere en eller flere af følgende: <ol style="list-style-type: none"> a. Brugernavn b. E-mail c. Password 2. Aktøren klikker "Bekræft" | | |
| Variationer | - | | |
| Slutresultat | Succes | Alle oplysninger der ønskes ændret er blevet ændret | |
| | Fejl | <p>Oplysninger bliver ikke ændret efter aktør har trykket "bekræft"</p> <p>Aktøren har indtastet en ugyldig e-mail</p> <p>Aktøren har indtastet en e-mail der i forvejen bliver brugt til en anden bruger</p> <p>Aktøren har indtastet et ugyldigt password</p> | |

| | | | |
|--------------------|---|---|--------------|
| Use Case ID: | A.4 - Slet bruger | | |
| Mål | Bruger bliver succesfuldt slettet | | |
| Initiering | Initieres af bruger | | |
| Aktør(er) | Bruger | | |
| Type | Selvstændig: X | Include: | Extends: A.3 |
| Frekvens | 10 - 50 om ugen | Samtidige forekomster: | 1 - 5 |
| Ikke-funktionskrav | Bruger bliver sendt tilbage til start/log-ind skærmen | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har en bruger i forvejen og er inde på "Rediger bruger" | | |
| Beskrivelse | 1. Bruger klikker på "Slet bruger" 2. Bruger klikker "ja" på sikkerheds pop-up | | |
| Variationer | - | | |
| Slutresultat | Succes | Brugerens konto og data bliver fjernet fra database | |
| | Fejl | Brugerens konto bliver ikke fjernet fra database Brugerens data bliver ikke fjernet fra database | |

| | | | |
|--------------------|--|---|----------|
| Use Case ID: | A.5 - Authorization | | |
| Mål | Gæst bliver succesfuldt logget ind som bruger | | |
| Initiering | Initieres af Use case "A.3. - Log ind" | | |
| Aktør(er) | API | | |
| Type | Selvstændig: X | Include: | Extends: |
| Frekvens | 10 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har en bruger i forvejen API og database serverne er oppe og køre | | |
| Beskrivelse | 1. Aktør modtager request til at logge ind 2. Aktør verificere oplysningerne i requesten 3. Aktør laver et reponse tilbage til applikationen | | |
| Variationer | - | | |
| Slutresultat | Succes | Aktør sender et godkendt reponse tilbage til applikationen Aktør sender "fejl" tilbage til applikationen | |
| | Fejl | Aktør kan ikke godkende requesten | |

| | | | |
|--------------------|--|--|----------|
| Use Case ID: | B.1 - Scan næringsindhold | | |
| Mål | Bruger får en fødevare tilføjet ud fra en scanning | | |
| Initiering | Initieres af Bruger | | |
| Aktør(er) | Bruger | | |
| Type | Selvstændig: X | Include: B.7 | Extends: |
| Frekvens | 1 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | Bruger kan se produktet i fødevare listen med det samme | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har åbnet appen og trykket på "Scan" | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktøren har scannet næringsindhold af et produkt 2. Aktøren får næringsindholdet tastet ind for sig 3. Aktøren klikker godkend Extend: Use case "B.4. - Godkend produkt" initieres <ol style="list-style-type: none"> 1. Alle brugere kan se fødevaren | | |
| Variationer | Extend: Use case "B.4. - Godkend produkt" <ol style="list-style-type: none"> 1. Fødevare bliver afvist 2. Aktøren kan kun selv se fødevaren | | |
| Slutresultat | Succes | Aktøren kan se sit eget fødevareprodukt | |
| | Fejl | Næringsindholdet kan ikke scannes ordentligt og der vises fejlbesked | |

| | | | |
|--------------------|---|--|----------|
| Use Case ID: | B.2 - Indtast fødevare manuelt | | |
| Mål | Fødevarer bliver indtastet manuelt og tilføjet til database | | |
| Initiering | Initieres af bruger | | |
| Aktør(er) | Bruger | | |
| Type | Selvstændig: X | Include: B.7 | Extends: |
| Frekvens | 1 - 1000 om ugen | Samtidige forekomster: | 1 - 50 |
| Ikke-funktionskrav | Bruger kan se produktet i fødevare listen med det samme | | |
| Bemærkninger | Fødevaren bliver ikke oprettet med en streghkode | | |
| Startbetingelser | Aktøren er logget ind og har trykket på "indtast manuelt" | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktøren klikker ind på "ny kostindtag" 2. Aktøren indtaster navn på fødevaren 3. Aktøren indtaster næringsindhold for fødevaren 4. Aktøren indtaster hvor meget fødevaren vejer 5. Aktøren trykker "godkend" | | |
| Variationer | - | | |
| Slutresultat | Succes | Alle oplysninger om fødevaren er indtastet og sendt ind i databasen | |
| | Fejl | <p>Oplysningerne bliver ikke sendt ind efter aktør har trykket "bekræft"</p> <p>Aktøren har ikke udfyldt alle felter</p> | |

| | | | |
|--------------------|---|---|----------|
| Use Case ID: | B.3 - Tilføj fødevare | | |
| Mål | Admin tilføjer et produkt alle kan tilføje til deres kost | | |
| Initiering | Initieres af Administrator | | |
| Aktør(er) | Administrator | | |
| Type | Selvstændig: X | Include: B.7 | Extends: |
| Frekvens | 1 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | Fødevaren bliver tilgængelig for alle brugere | | |
| Bemærkninger | - | | |
| Startbetingelser | Aktøren har åbnet appen og tilgået admin panelet | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktøren klikker tilføj nyt fødevareprodukt 2. Aktøren skanner en strejkode eller indtaster næringsindhold manuelt 3. Aktøren klikker opret 4. Fødevaren bliver registreret på databasen | | |
| Variationer | - | | |
| Slutresultat | Succes | - Brugere kan se fødevare produktet og tilføje det til deres kost | |
| | Fejl | Aktør har ikke udfyldt felterne ordentligt Oplysningerne bliver ikke sendt ind efter aktør har klikket "opret" | |

| | | | |
|--------------------|--|---|----------|
| Use Case ID: | B.4 - Godkend fødevare | | |
| Mål | Fødevare bliver godkendt af admin | | |
| Initiering | Initieres af administrator | | |
| Aktør(er) | Administrator | | |
| Type | Selvstændig: X | Include: | Extends: |
| Frekvens | 1 - 5000 om ugen | Samtidige forekomster: | 1-50 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | Alle fødevarer skal godkendes manuelt af administrator | | |
| Startbetingelser | Aktøren er logget ind og er inde på "godkend fødevarer" listen | | |
| Beskrivelse | 1. Aktøren klikker ind på fødevarer 2. Aktøren trykker "godkend" 3. Aktøren fortsætter for alle fødevarer der skal godkendes | | |
| Variationer | 2.a Aktøren trykker fortryd 3.a Fødevaren kan ikke søges på af andre brugere | | |
| Slutresultat | Succes | Alle fødevare er godkendt succesfuldt og kan tilgås af alle brugere | |
| | Fejl | Fødevarer bliver ikke opdateret med "godkendt" status | |

| | | | |
|--------------------|--|--|----------|
| Use Case ID: | B.5 - Rediger fødevare | | |
| Mål | Fødevare bliver succesfuldt redigeret | | |
| Initiering | Initieres af administrator | | |
| Aktør(er) | Administrator | | |
| Type | Selvstændig: X | Include: | Extends: |
| Frekvens | 1 - 100 om ugen | Samtidige forekomster: | 1-10 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | Fødevarer kan kun redigeres af en bruger med admin rolle | | |
| Startbetingelser | Aktøren er logget ind som admin | | |
| Beskrivelse | <ol style="list-style-type: none"> 1. Aktøren trykker ind på en fødevare 2. Aktøren trykker "Rediger" 3. Aktøren kan nu redigere følgende: <ol style="list-style-type: none"> a. Fødevarens navn b. Fødevarens næringsindhold c. Fødevarens godkendelse 4. Aktøren trykker "Godkend" | | |
| Variationer | - | | |
| Slutresultat | Succes | Alle fødevare er succesfuldt redigeret | |
| | Fejl | Fødevarer bliver ikke opdateret med de nye informationer | |

| | | | |
|--------------------|---|---|--------------|
| Use Case ID: | B.6 - Slet fødevare | | |
| Mål | Fødevare bliver godkendt af admin | | |
| Initiering | Initieres af administrator | | |
| Aktør(er) | Administrator | | |
| Type | Selvstændig: | Include: | Extends: B.5 |
| Frekvens | 1 - 100 om ugen | Samtidige forekomster: | 1-10 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | Fødevarer kan kun slettes af en bruger med admin rolle | | |
| Startbetingelser | Aktøren er logget ind som admin og er under "Rediger" skærmen | | |
| Beskrivelse | 1. Aktør trykker "Slet" 2. På pop-op'en trykker aktøren "Ja" | | |
| Variationer | 2.a Aktøren trykker "Nej" | | |
| Slutresultat | Succes | Fødevaren er succesfuldt slettet fra databasen | |
| | Fejl | Fødevareren er ikke slettet og kan stadig søges på / ses i database | |

| | | | |
|--------------------|---|---|----------|
| Use Case ID: | B.7 - Tilføj fødevare i database | | |
| Mål | Fødevare bliver registreret som godkendt i databasen | | |
| Initiering | Admin tilføjer eller godkender et fødevareprodukt | | |
| Aktør(er) | API | | |
| Type | Selvstændig: | Include: | Extends: |
| Frekvens | 1 - 100 om ugen | Samtidige forekomster: | 1 - 10 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | - | | |
| Startbetingelser | API samt database serveren kører | | |
| Beskrivelse | 1. Fødevare bliver registreret i databasen | | |
| Variationer | 1.a Fødevare bliver registreret som "ikke godkendt" i databasen | | |
| Slutresultat | Succes | Registreringen kan ses i databasen | |
| | Fejl | Registreringen bliver ikke oprettet i databasen | |

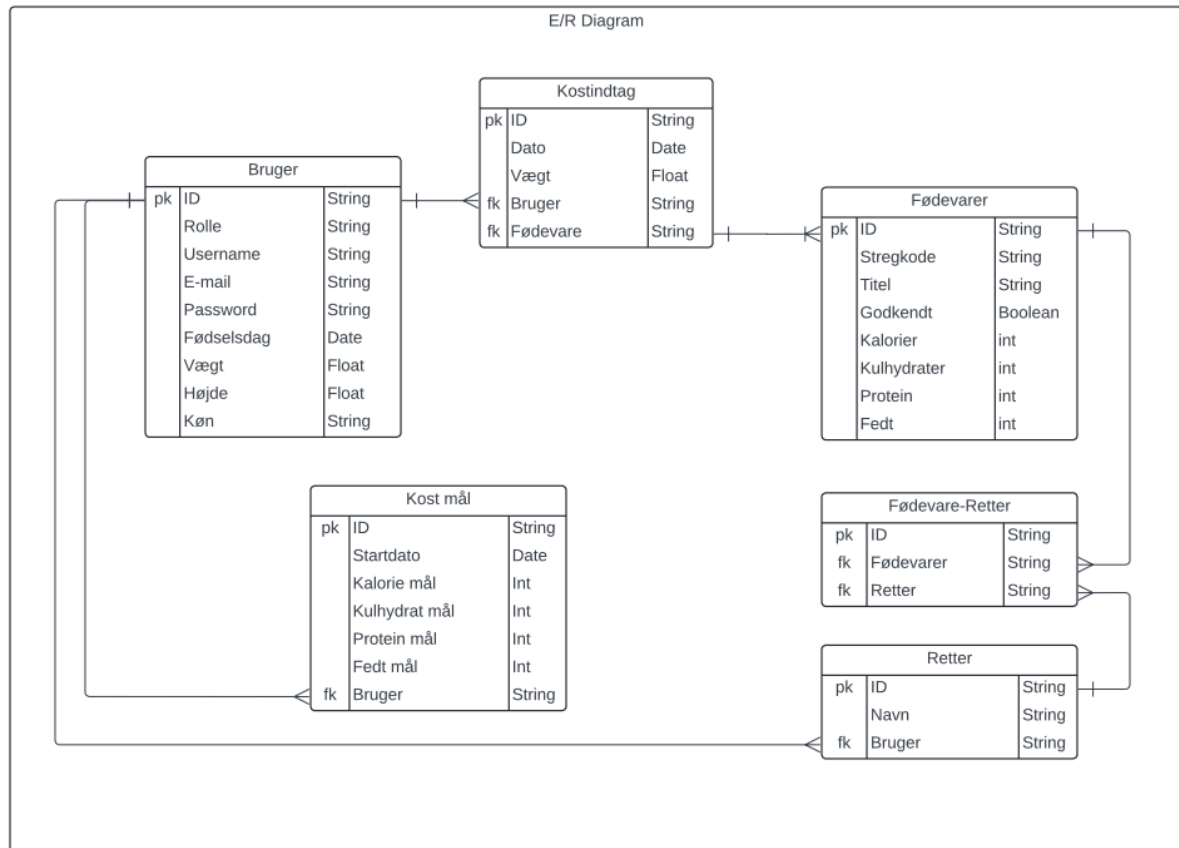
| | | | |
|--------------------|--|--|------------|
| Use Case ID: | C.1. - Anmod tracking data | | |
| Mål | Data til makrotracking bliver vist til brugeren | | |
| Initiering | Bliver automatisk anmodet når brugeren er på dashboardet | | |
| Aktør(er) | Bruger | | |
| Type | Selvstændig: X | Include: C.2. | Extends: - |
| Frekvens | 1 - 10000 om ugen | Samtidige forekomster: | 1 - 500 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | Tracking processen sker automatisk på frontend med udregninger ud fra den anmodet data | | |
| Startbetingelser | - | | |
| Beskrivelse | 1. Aktøren dirigeres / trykker ind på dashboardet 2. C.2. Initieres | | |
| Variationer | - | | |
| Slutresultat | Succes | Aktøren kan se det tracket data | |
| | Fejl | Data bliver ikke vist / kun delvist vist | |

| | | | |
|--------------------|--|---|----------|
| Use Case ID: | C.2. - Hent tracking data | | |
| Mål | Data til makrotracking bliver hentet ned fra databasen | | |
| Initiering | Når bruger anmoder om tracking data (C.1.) | | |
| Aktør(er) | API | | |
| Type | Selvstændig: | Include: C.1. | Extends: |
| Frekvens | 1 - 10000 om ugen | Samtidige forekomster: | 1 - 500 |
| Ikke-funktionskrav | - | | |
| Bemærkninger | - | | |
| Startbetingelser | API og Database server er oppe og køre | | |
| Beskrivelse | 1. Aktøren henter data ned fra database 2. Aktøren sender dataen til frontend | | |
| Variationer | - | | |
| Slutresultat | Succes | Data bliver succesfuldt sendt til frontend | |
| | Fejl | Data bliver ikke / kun delvist sendt til frontend | |

E/R Diagram

E/R diagrammet viser, hvad databasen skal indeholde af tabeller samt deres forhold til hinanden.

Brugers indtaget kalorier vil blive afregnet ved at hente alle ting i "kostindtag" filtreret efter den givne dag.



Mockup

Mockup'et findes i vores github repository under Bilag/Mockup/Fuld mockup.png

Guide

Mockup'et har funktionen af et flowchart.

Farvernes betydning:

Rød: Admin

Lilla: Opret ny bruger

Grøn: Bruger tjekker log eller opdatere indstillinger

Blå: Bruger tilføjer en enkelt vare

Orange: Bruger tilføjer en ret

Sort: ekstra ting

En mere detaljeret beskrivelse står ved siden af pilene.

Tidsplan

Den ønskede tidsplan vi håber vi kan følge:

