Kravspecifikation

**Tilføjelse af en cloud baseret løsning og api til håndtering af Favoritter og Login**

**Kravspecifikation indhold**

Indholdsfortegnelse

[Ledelsesresume 3](#_Toc58401369)

[Baggrund 3](#_Toc58401370)

[Situation 3](#_Toc58401371)

[Problemløsning 3](#_Toc58401372)

[Overblik 4](#_Toc58401373)

[Løsningsskitse 4](#_Toc58401374)

[Mål og kritiske succesfaktorer 4](#_Toc58401375)

[Nuværende løsning og problemformulering 5](#_Toc58401376)

[Fremtidig løsning 6](#_Toc58401377)

[2nd Iteration 6](#_Toc58401378)

[Konsekvenser for virksomheden 7](#_Toc58401379)

[Implementeringskrav 7](#_Toc58401380)

[Effektivitetsmodel 7](#_Toc58401381)

[Software Sikkerhed 8](#_Toc58401382)

[Funktionelle Krav 9](#_Toc58401383)

[Kontekst 9](#_Toc58401384)

[Overordnede krav og succeskriterier 10](#_Toc58401385)

[Informationsmodel 11](#_Toc58401386)

[Procesmodel 11](#_Toc58401387)

[Prototype 12](#_Toc58401388)

[Database 12](#_Toc58401389)

[Database Sikkerhed 12](#_Toc58401390)

[Løsningsarkitektur og leveranceopdeling 13](#_Toc58401391)

[Prioriterede forretningskrav 14](#_Toc58401392)

[Use cases 14](#_Toc58401393)

[Testsituationer og acceptkriterier 15](#_Toc58401394)

[Ikke funktionelle krav 17](#_Toc58401395)

[Krav til virksomheden 17](#_Toc58401396)

[Fysiske rammer 17](#_Toc58401397)

[Hardware/softwaremiljø 17](#_Toc58401398)

[Krav til systemegenskaber 17](#_Toc58401399)

# Ledelsesresume

Virksomheden ønsker at få udviklet et login system til deres Radio applikation.  
Som gøre det muligt at gemme sine favoritter og indstillinger mellem telefoner

# Baggrund

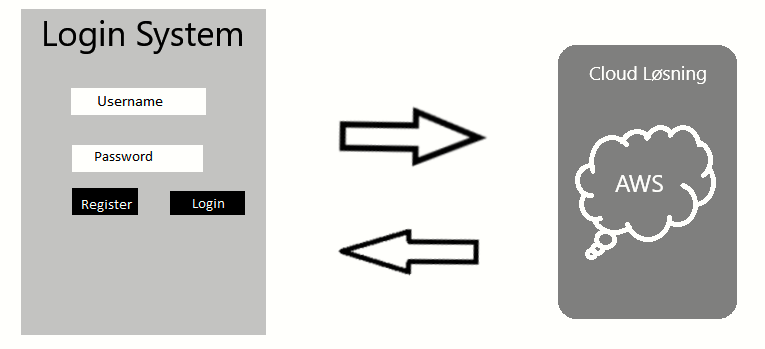
## Situation

Brugerne er trætte af at hente alle favoritterne igen enkeltvis, hvis de får en ny mobil telefon eller bruger flere.  
Applikationens brugere, som får en ny telefon eller factory reset telefonen mister de dermed alle deres gemte indstillinger og favoritter, da favorit listen og indstillinger gemmes på selve mobilen.

## Problemløsning

Der skal videreudvikles en cloud baseret løsning og et login system i appen. Der bruges en cloud baseret løsning til at centralisere alt data et sted og ved hjælp af internet kan bruge løsningen. Cloud løsningen skal indeholde en API, som skal håndterer alle requests fra brugeren og databasen .Login system skal bruges til at identificerer brugeren og hente de rigtige data ned fra databasen. Løsningen skal erstatte den nuværende løsning og gemme alle relevante data på en centraliseret database. Og dermed kan man logge ind på appen og få sine gemte favoritter og indstillinger. Opgaven skal løses iterativt og færdigudvikles. Applikationen designes og dokumenteres som en løsning på en mobil.

# Overblik

Løsningsskitse **Figur 1**  
****

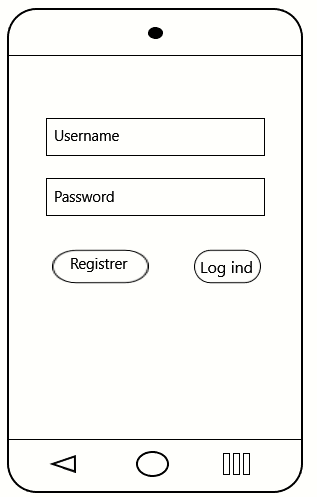
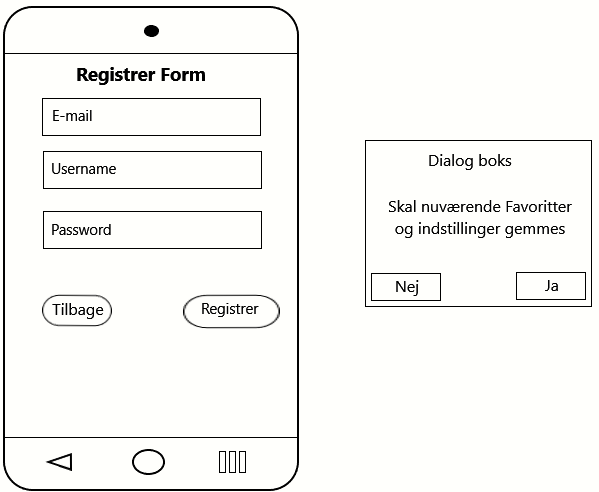
Mål og kritiske succesfaktorerInteressenter og deres kritiske succesfaktorer beskrives som følger:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Interessent** | **Projektmål** | **KSF**  **(Kritisk Succes Faktor)** | **Adresseret i kravspecifikation**  **(0 – 100%)** | **Begrundelse for under 100% adressering af KSF** |
| Bruger | Intuitivt | At login systemet er minimalistisk | 100% |  |
| Database | Drift | at AWS-servicen aldrig bliver slukket for og altid adgang til databasen | 100% |  |
| Database | Migration | Migrer alle data fra den lokale database til Cloud løsningen | 100% |  |
| Login System | Funktioner | At man kan logge ind og registrer med brug af Sikkerhedskravene | 90% | At login og registrer funktioner kan anvendes. |
| Applikation | Sikkerhed | Sikkerhed imod brute force, sql injection og sanitizing | 100% |  |
| Bruger | Glemt adgangskode Funktion | Man kan reset sin adgangskode via et link man får tilsendt sin e-mail | 20% | At funktionen er der og præsenteres, men kan ikke anvendes. |
| Applikation | Internet/Wifi | Applikationen skal have adgang til internettet | 100% |  |
| Bruger | Konto | En Bruger kan lave flere kontoer med samme e-mail | 100% |  |
| Database | Konto | Et brugernavn er unikt og adgangskode kan variere. | 100% |  |
| API | Authentication | Authentication til at authenticate applikationen/enheden for at kunne bruge apien | 100% |  |
| API | Forbindelse | Forbinde databasen med apien og CRUD funktioner | 90% | Forbindelse og alle CRUD undtagen update virker og anvendes |
| Applikation | Forbindelse | Forbinde applikationen med apien | 100% |  |

Nuværende løsning og problemformuleringI dag gemmer brugerne deres favoritter og indstillinger på local storage, som ligger i deres egne telefoner. Hvis brugeren får en ny telefon, kan den ikke overføre deres favoritter og indstillinger til deres nye telefon og bruger derfor for lang tid til at tilpasse deres nye telefon.

Fremtidig løsningDer skal videreudvikles et login system og migrerer den lokale database over til cloud løsningen (Ekstern database), hvis egenskaber er, at man kan logge ind og identificere personen, derefter gemme eller hente favoritter og indstillinger knyttet til personen fra eller til cloud (Ekstern database). Cloud løsningen er baseret på AWS (Amazon Web Services) med knytning til MySQL Database. Migrer brugernes data fra den lokale database til cloud løsningen. Med en dialog boks funktion, som spørger brugeren efter at han/hun er registreret om nuværende data skal gemmes i cloud.Adgangskode skal kypteres og Email skal valideres for at kunne registrer en konto. Dialog boksen bliver kun supportet i de første 3 måneder, derefter vil appen blive opdateret uden funktionen. Hvis brugeren siger nej til dialog boksen vil kontoen blive oprettet som ny. Login System skal være minimalistisk i design og brugervenlig i synlighed. Som kan ses på Figur 1 og 2. Hele løsningen bliver baseret på **3 layer application model**. Databasen skal via en api have tilknytning til applikationen

**Figur 2**

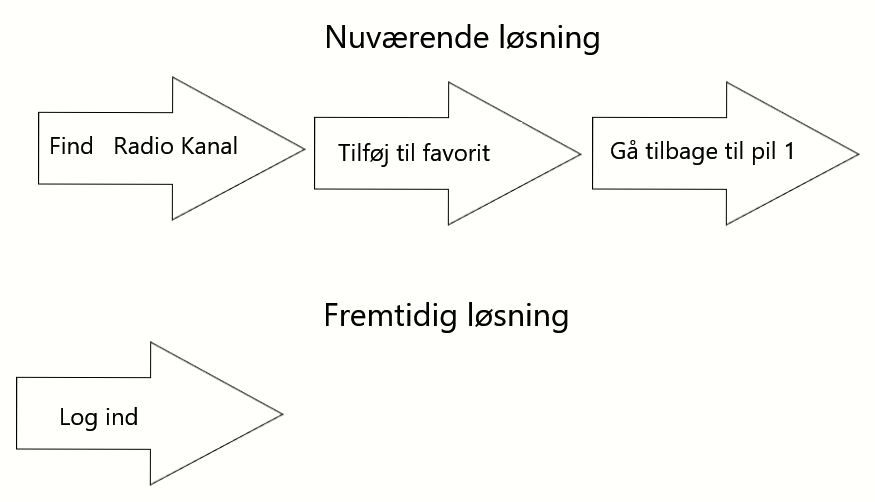
****  
  
  
2nd Iteration  
I anden iteration skal authentication laves på apien, som skal sikre at applikationen/enheden er godkendt til at bruge apien. Funktionen glemt adgangskode laves, så vil man via applikaitonen blive sendt en mail til kontoens e-mail address og i mailen vil man få et link til at ændre sin adgangskode. Der vil blive tilføjet andre private indstillinger end nuværende, hvor der kun er en Favoritliste.

Konsekvenser for virksomhedenApplikationens brugere køber nye telefoner, bruger flere telefoner eller låner en anden persons telefon. Med denne løsning kan man tage sine favoritter og indstillinger med hvor og når som helst. Så længe brugeren logger ind med sin egen konto på applikationen. Brugerne vil spare tid med at finde sin favorit radiokanal på en anden telefon og spare tid med at flytte sin gamle favorit liste over på den nye telefon. Cloud løsningen vil også sikre en fremtidig fremgangsmåde og sikre applikationen flere udviklingsmuligheder i fremtiden, hvis der bliver behøv for det.

ImplementeringskravLeverandøren sørger for at have databasen og apien kørende på AWS og regelmæssigt tager backup af databasen. Leverandøren skal afholde et workshop for virksomhedens IT ansvarlige i anvendelse af applikationen, samt et gennemgang af koden bag applikationen, så Virksomheden selv kan udvikle og rette på applikationen. Virksomheden står selv for at uddanne de medarbejdere der skal anvende applikationen. Applikationen skal have adgang til WIFI/Internet.

## Effektivitetsmodel

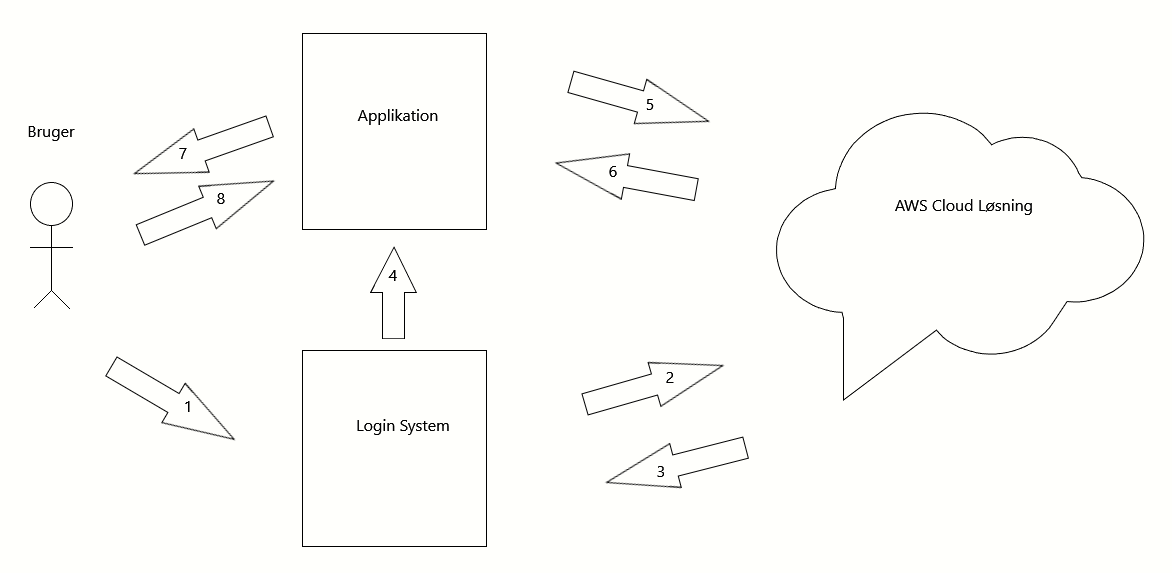
Her kan man se at med den nuværende løsning kræver det 2 steps for en bruger med en ny telefon for at tilføje en kanal til sin favorit liste. Men det er kun i tilfælde af 1 tilføjelse. Hvis brugeren havde flere favoritter på sin gamle telefon, som han skal overføre til sin nye, kræver det step **Find Radio Kanal + Tilføj til favorit \* N**, hvor **N** er antallet af favoritter brugeren skal tilføje til sin nye telefon. Nu bliver den nuværende løsning mere tidskrævende end nogensinde. Med Leverandørens beregninger vil det som minimum tage 10 sekunder for et cyklus dvs. at det tager 10 sekunder for at finde og tilføje **1** kanal til favorit.  
Med den fremtidig løsning kræver det kun 1 step. Man logger ind med sin bruger konto. Dermed henter applikationen alle brugerens gemte favoritter fra databasen. Dette vil tager op til 20 sekunder. Men dette skal man kun gøre 1 gang.



Software SikkerhedAf sikkerhedsmæssige grunde bliver adgangskoder og e-mail **krypteret**, når man bliver registreret og **AWS best practice configuration** skal bruges, som en del af cloud database sikkerheden.  
Alle inputs felter vil blive desinficeret for **SQL injektion** og login systemet skal beskyttes imod brute force attack. Der skal bruges TLS på alt kommunikation på applikationen.

* Kryptering af adgangskoder på cloud databasen
  + adgangskoden bliver sha384 krypteret med BCrypt krypteringsalgoritmen et tilfældig genereret salt på 16 byte
  + Både krypteret adgangskode og saltet bliver gemt på cloud databasen.
  + Leverandøren har valgt en slow hash function for at beskytte imod lookup tables og brute force. Det gør det langsommere for hackers at brute force, ved at have en større hash og at det tager lidt tid at dekryptere det. Det påvirker ikke performance, jo mindre man prøver at brute force sig ind med flere forsøg.
* AWS sikkerhedsstandarder
  + Der bliver kørt standard AWS sikkerhed.
  + Der bliver lavet 2 end points. Reader og Writer end points. Writer er til at skrive data ind på databasen og Reader er for at læse fra databasen. Som skal bruges til at kommunikere med databasen
* Alle inputs felter skal desinficeret og beskyttet fra SQL injektioner
  + Desinficering sker i backend med prepared statements for at beskytte imod mulige angreb og SQL Injektioner.
* Login systemet skal beskyttes imod brute force ved at give brugeren 3 login forsøg, derefter vil kontoen ud fra brugernavnet blive deaktiveret og en mail sendt ud. Kontoen kan kun blive genaktiveret på den sendte mail.
* Applikationen skal bruge TLS 1.2+ protokollen for alt ekstern kommunikation
* Ved hjælp af API skal enhedens blive godkendt af apien for at bruge den. Det sker ved hjælp at logge ind på applikationen på enheden.

Funktionelle Krav

****KontekstFiguren viser overordnet hvordan løsningen fungerer **Figur 3**

De enkelte interessenter beskrives som nedenfor angivet:  
  
Rolle: Bruger **Pil 1**  
Data: Brugeren indtaster sit brugernavn og adgangskode.  
  
Rolle: Login System **Pil 2 og 3**  
Data: Login systemet tjekker om brugernavn og adgangskode passer med en konto fra cloud databasen.

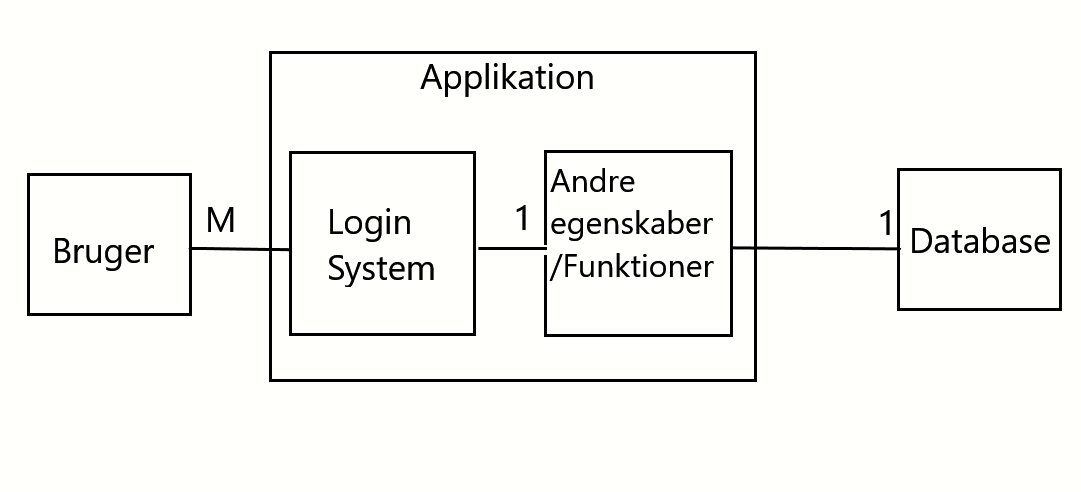
Rolle: Login System **Pil 4**  
Data: Hvis brugernavn og adgangskode passer, så navigerer der til applikationen.  
  
Rolle: AWS Cloud Løsning (Ekstern Database) **Pil 2,3 og 5,6**  
Data: Centralisere alle kontoer og tilknyttet favoritter, samt indstillinger på en database. Håndtering af data.  
  
Rolle: Applikation **Pil 5 og 6**  
Data: Henter all favoritter og indstillinger tilknyttet kontoen, som er blevet brugt til at logge ind. Og fremviser det på applikationen, som favorit liste.  
  
Rolle: Applikation **Pil 7 og 8**  
Data Fremviser radio applikation view til brugeren med alle nødvendige data. Brugeren kan nu bruge applikationen **Pil 5,6 og 7,8** uden at gå igennem **Pil 1, 2 og 3** igen jo mindre man har logget ud.

Overordnede krav og succeskriterier

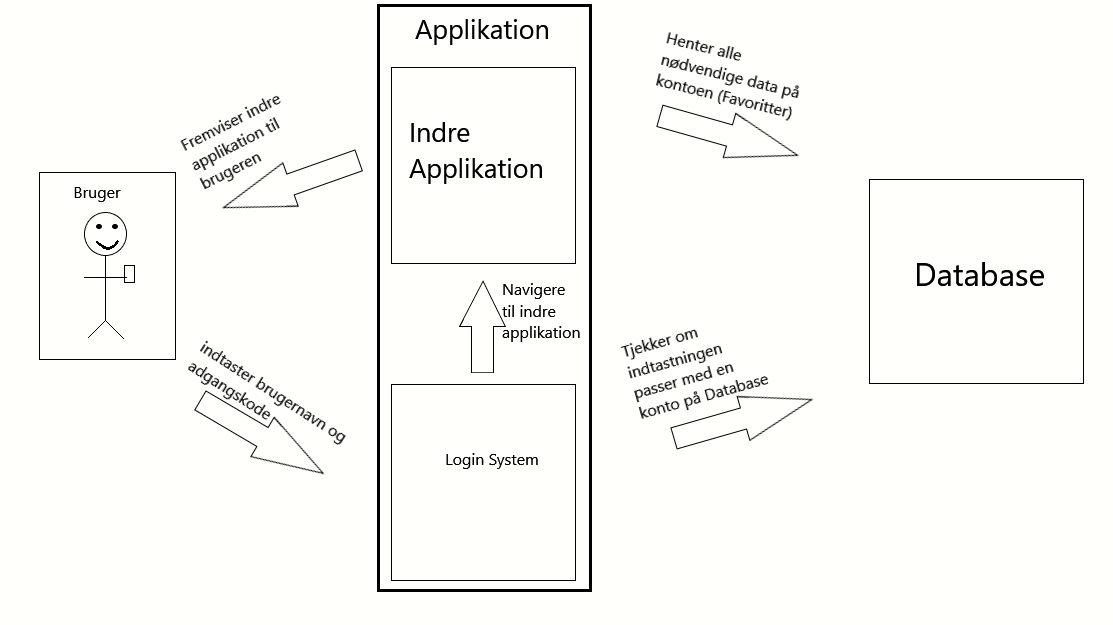
|  |  |  |  |
| --- | --- | --- | --- |
| **Interessent** | **Projektmål** | **KSF (Kritisk Succes Faktor)** | **Overordnet krav** |
| Bruger | Intuitivt | Det er vigtigt at Login system kan anvendes uden behov for hjælp | Applikationen skal have en moderne og minimalistisk design |
| Database | Centralisering af Data | Centralisere favoritter, kontoer og andre vigtige data | Dataene er tilknyttet en konto, så hver konto har deres egne data |
| Applikation | Login System | Man kan logge ind og registrer sig | Oprette en ny konto på databasen og matche data mellem app og database |
| API | Forbindelse til Database og sikkerhed | JWT tokens og forbindelse til Database og CRUD funktioner | At apien bruger CRUD for at forbinde til databasen. Med jwt tokens for at authenticate brugeren |
| Applikation | Favoritter | Man kan hente Brugerens favorit liste og indstillinger fra cloud databasen | Hente kontoens data fra databasen med brug af konto ID |
| Applikation | Tilføj/slet favorit | Man skal kunne tilføje og slette favoritter | Man skal kunne tilføje og slette favoritter fra favorit listen og fra databasen tilknyttet kontoen |

Informationsmodel

Denne figur beskriver at der vil være mange brugere som anvender applikationen, men brugeren skal først igennem login systemet før de bliver navigeret videre ind i applikationen. Applikationen er tilknyttet 1 database.

**Figur 4**

ProcesmodelFunktion analyse data flow diagram  
 **Figur 5**

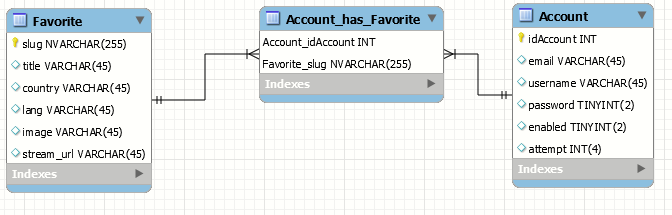


Prototype

Se løsningsskitserne figur 1 og 2.

Database

Her kan du visuelt se database modellen. Account kan have flere favoritter og Favoritter kan have flere Accounts tilknyttet. Leverandøren vil bruge mysql database til løsningen.   
 **Figur 6**



Database SikkerhedLeverandøren står for backup af databasen regelmæssigt, men gemmer kun de 3 sidste backups. Virksomhedens database administratorer skal selv sørge for at tage en backup, hvis virksomheden vil gemme tidligere backups. Leverandøren laver 2 endpoints i databasen. Reader og Writer. Reader er til at se og hente data fra databasen og Writer er for at skrive til databasen. Dette skal sikre imod unødvendig rettigheder og tilladelser.

## Løsningsarkitektur og leveranceopdeling

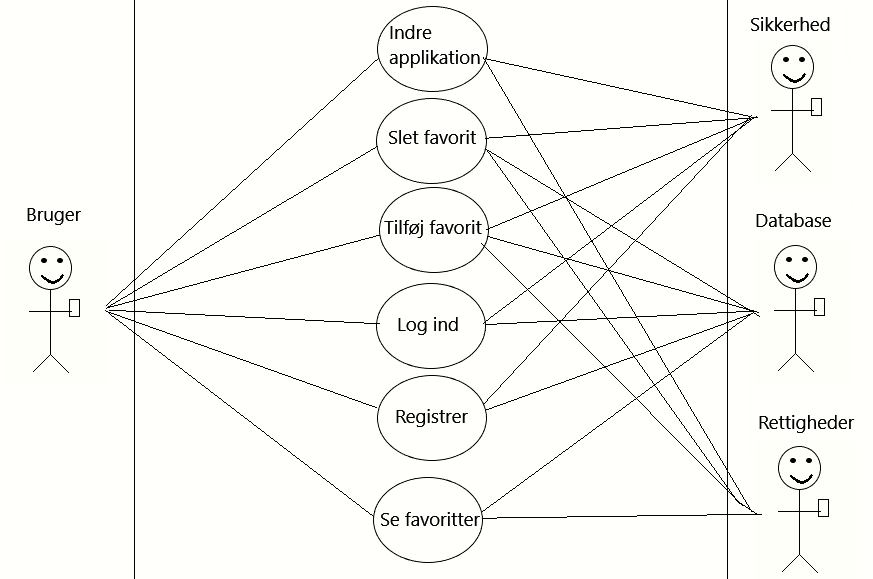
Her er beskrevet hvordan den samlede løsning skal leveres. X er startdato for opgavens påbegyndelse.

1. **X + 1 uge**
   1. Levering af graphic user interface GUI på registrering og login skærmene uden virkende funktioner
   2. Virksomheden godkender den grafiske brugerflade
   3. Dette har prioritet 1
2. **x + 2 uger**
   1. Leverandøren sætter, opretter og aktiver cloud databasen på AWS
   2. Virksomheden i samarbejde med leverandøren opretter SQL tabellerne og kontrollerer at tabellerne er skrevet rigtigt ind
   3. Dette har prioritet 2
3. **x + 3 uger**
   1. Leverandøren sætter apien op med de aftalte database og CRUD funktioner
   2. Leverandøren sætter authentication op
   3. Virksomheden kontrollerer at de kan koble til apien og bruge databasen via apien, samt kontrollerer at authentication fungerer.
4. **x + 4 uger**
   1. Leverandøren tilføjer login funktionen til applikationen
   2. Leverandøren tilføjer navigation til registrer funktionen til applikationen
   3. Virksomheden kontrollerer at et korrekt brugernavn og adgangskode passer sammen med en konto fra cloud databasen og logger ind
   4. Virksomheden kontrollerer at et forkert brugernavn og adgangskode ikke passer sammen med en konto fra cloud databasen og skærmen vises med skrift ”forkert brugernavn eller adgangskode”
   5. Virksomheden kontrollerer at navigation til registrer skærmen virker
   6. Dette har prioritet 3
5. **x + 5 uger**
   1. Leverandøren tilføjer registrer funktionen til applikationen
   2. Leverandøren tilføjer tilbage funktionen til applikationen
   3. Virksomheden kontrollerer at ved registreringen bliver dataene sendt over til cloud databasen
   4. Virksomheden kontrollerer at tilbage funktionen virker.
   5. Dette har prioritet 3
6. **x + 6 uger**
   1. Leverandøren tilføjer at de korrekte favoritter bliver hentet ned fra cloud databasen og fremvises på applikationen.
   2. Virksomheden kontrollerer at på test kontoer ses de rigtige favoritter
   3. Dette har prioritet 4
7. **x + 7 uger**
   1. Leverandøren tilføjer tilføj favorit funktionen til applikationen
   2. Virksomheden kontrollerer at den tilføjet favorit bliver sendt over til cloud database med den rigtige konto ID tilknyttet.
   3. Dette har prioritet 4
8. **x + 8 uger**
   1. Leverandøren tilføjer slet favorit funktionen til applikationen
   2. Virksomheden kontrollerer at den slettede favorit er slettet eller deaktiveret i databasen på den korrekte konto ID
   3. Dette har prioritet 4
9. **x + 9 uger**
   1. Leverandøren sikkerhedsklargør funktionerne som er beskrevet i **3.7**
   2. Virksomheden kontrollerer at sikkerheden virker
   3. Dette har prioritet 5

## Prioriterede forretningskrav

Se overstående.

## Use cases

 **Figur 7**

## Testsituationer og acceptkriterier

* 1. Der er indlagt FAT( factory acceptance test) i forbindelse med del leverancer se i afsnittet Løsningsarkitektur og leveranceopdeling.SAT (site acceptance test) gennemføres ved komplet leverance.

**Vilkår for SAT gennemførelse:**

* Hvis der opstår en fejl i applikationen, har leverandøren 5 timer til at forsøge at rette problemet, hvis fejlen fremgår i leverandørens kode eller database. Virksomhedens SAT ansvarlige tilkalder leverandørens programmør ved fejl i applikationen
* Leverandøren har en programmør på kald, da leverandøren ikke selv deltager i SAT. Denne programmør skal kunne tilkaldes af SAT ansvarlig
* Virksomheden har inden test gjort applikationen tilgængelig på de relevante test mobiler.

**Under SAT gennemføres følgende test:**

1. På en klient mobil startes applikationen  
   **1.1** Der kontrolleres at der er forbindelse til cloud databasen  
   **1.2** Succes: Der er forbindelse til cloud databasen  
   **1.3** Fejl: Der er ikke forbindelse til cloud databasen
2. Først registrer man en ny konto  
   **2.1** på log ind skærmen vælger man registrer  
    **2.1.1** Succes: man bliver navigeret til registrerings skærmen  
    **2.1.2** Fejl: man er stadig på log ind skærmen  
   **2.2** Man indtaster oplysninger i input felterne (brugernavn, e-mail, adgangskode)  
   **2.3** Man trykker på registrer knappen  
    **2.3.1** Succes: Man bliver navigeret tilbage til log ind skærmen  
    **2.3.2** Fejl: Man er stadig på registrer skærmen  
   **2.4** Der bliver kontrolleret på cloud databasen at kontoen er oprettet  
    **2.4.1** Succes: kontoen er oprettet på konto tabellen  
    **2.4.2** Fejl: Kontoen kan ikke ses på konto tabellen
3. Dernæst logger man ind  
   **2.1** Der kontrolleres at log ind knappen er aktiv  
   **2.2** Man indtaster et korrekt brugernavn og adgangskode fra en registreret konto og trykker på log ind  
    **2.3.1** Succes: Man er navigeret til indre applikation  
    **2.3.2** Fejl: Man er stadig på log ind skærmen
4. Dernæst kontrolleres at kontoen har fået de rigtige favoritter  
   **4.1** Gå ind på favorit listen  
    **4.1.1** Succes: Man kan se alle de rigtige tilknyttet favoritter  
    **4.1.2** Fejl: Man kan ikke se nogen favoritter  
    **4.1.3** Fejl: Man kan kun se nogle af favoritterne  
    **4.1.4** Fejl: Man kan se favoritter fra andres kontoer
5. Dernæst kontrolleres at man kan tilføje en favorit til favorit listen  
   **5.1** Man trykker på en af kanalerne  
   **5.2** Man trykker på den sorte stjerne  
    **5.2.1** Succes: Stjernen bliver gult  
    **5.2.2** Fejl: Stjernen er fortsat sort  
   **5.3** Kontrollerer at favoritten er nu tilføjet til favorit listen  
    **5.3.1** Gå ind på favorit listen  
    **5.3.1.1** Succes: Favoritten er blevet tilføjet til favorit listen  
    **5.3.1.2** Fejl: Favoritten kan ikke ses på favorit listen  
   **5.4** Kontrollerer at favoritten er nu tilføjet til cloud databasen  
    **5.4.1** Gå ind på cloud databasen og ind på Favorit tabellen og tjekke om favoritten er oprettet og tilknyttet konto ID’en  
    **5.4.2.1** Succes: Favoritten kan ses på Favorit tabellen med den rigtige konto id  
    **5.4.2.2** Fejl: Favoritten er ikke oprettet på Favorit tabellen  
    **5.4.2.3** Fejl: Favoritten er oprettet, men med forkert konto ID
6. Til sidst kontrolleres at man kan slette en favorit fra favorit listen  
   **6.1** Tryk på en af kanalerne fra favorit listen  
   **6.2** Tryk på den gule stjerne  
    **6.2.1** Succes: Den gule stjerne bliver sort  
    **6.2.2** Fejl: Den gule stjerne ændre sig ikke  
   **6.3** Kontrollerer at favoritten er blevet fjernet fra favorit listen  
    **6.3.1** Gå ind på favorit listen  
    **6.3.1.1** Succes: Favoritten kan ikke længere ses på favorit listen  
    **6.3.1.2** Fejl: Favoritten kan ses på favorit listen  
   **6.4** Kontrollerer at favoritten er blevet fjernet eller deaktiveret fra kontoen på cloud databasen  
    **6.4.1** Gå ind på cloud databasen og ind på favorit tabellen  
    **6.4.1.1** Succes: Favoritten er fjernet eller deaktiveret for den enkelte konto  
    **6.4.1.2** Fejl: Favoritten kan stadig ses uden ændringer  
    **6.4.1.2** Fejl: Favoritten kan stadig ses, men uden konto ID tilknyttet

For at SAT kan godkendes skal der opnås positivt resultat i punkterne 1-4.  
 Hvis der ikke opnår positivt resultat i punkt 5 og 6, skal disse udbedres i anden iteration.

Ikke funktionelle krav

Krav til virksomhedenVirksomheden stiller test personel, samt hardware til rådighed/software efter behov.

Fysiske rammer

Hardware/softwaremiljø

Softwaren er en overbygning på nuværende applikation. Applikationen skal anvende AWS cloud databasen og en api frem for den lokale database inde i telefonerne.

Krav til systemegenskaber

Systemegenskaber krav til:

* **Performance og kapacitet**

Svar tiden mellem database og klient må ikke overstige 5 sekunder

* **Pålidelighed og tilgængelighed**Virksomhedens IT-afdeling har ansvaret for driftsstabilitet af AWS cloud databasen.  
  Leverandøren har kun ansvaret for at have databasen kørende på AWS.
* **Vedligehold**Leverandørens kode bliver merget sammen med Virksomhedens eksisterende applikation på den valgte version styrings værktøj.  
  Virksomheden har muligheden for at videreudvikle og justere på applikationen.
* **Brugervenlighed**Det er vigtig at designet er intuitivt og minimalistisk opbygget. Det skal sikre mindre stress for øjnene og applikationen kan anvendes uden yderligere undervisning.
* **Skalering og flytbarhed**Cloud databasen skal sikre at man kan bygge applikationen på flere forskellige platforme og stadig bruge samme database, da dette vil sikre DAL lagen af 3 lags hierarki modellen. I anden iteration vil leverandøren udvikle et API til brug og færdiggøre 3 lags hierarki modellen.
* **Sikkerhed**Kun Virksomhedens database administratorer og leverandøren har adgang til databasen.