# Generelt koncept og de mest basale ting programmet skal kunne

## Generel beskrivelse

Et webbaseret system til at håndtere svømmestævner.

Tilmelding af svømmere Fordeling til heats ud fra tilmeldings tider og indskrivning af opnåede tider til stævnet samt resultatlister fordelt på heat og eller handicap både ud fra tider men også FINA point

Systemet skal endvidere kunne bruges til at følge en enkeltes svømmers fremgang / tider igennem årene

## Opret stævne:

Man skal kunne oprette et stævne med de nødvendige informationer 6-8 baner langkort bane nation sted dato region mm.

Der skal derudover være en liste med hvilke løb der er med til stævnet f.eks. 100 fri 200 fri

## Tilmeld svømmer:

(man skal) – Fremover vil en træner eller en klub kunne tilmelde en svømmer til et stævne i de løb svømmeren skal deltage i, tilmeldingen skal have de relevante data for det enkelte løb, handicap S6 SB7 (handicap klasser – (skal hentes fra svømmerens info men stadig skrives i tilmeldingen (Se datastruktur) samt tilmeldingstid, yderbane, 3 ure og andet relevant information for det givende løb)

## Oprettelse af heats:

Når tilmeldingsfristen er overstået skal der ud fra de tilmeldte svømmere i de enkelte løb fordeles heats således at hvis der i et løb er 11 tilmeldte og stævnet har 6 baner skal de 11 svømmere fordeles korrekt på banerne de hurtigste i midten de langsomste yderst med undtagelse af dem der har specielt krav om Yderbane.

Fordelingsalgoritmen skal der nok arbejdes lidt på f.eks. Vil vi ved 8 tilmeldinger hellere have 4 i 2 heats end 2 i første heat og 6 i det andet heat.

Det skal efter tilmelding være mulighed for at manuelt justere i heatene og være mulighed for at fjerne tilmeldte hvis de ikke dukker op til stævnet.

Stævne programmet skal kunne udskrives i et fornuftigt format hvor tidstagerne stadig kan notere tider ned og se de informationer der er nødvendige.

## Resultater:

Under stævnet skal resultaterne kunne skrives ind og resultatlister blive lavet og udskrevet. Resultatlister skal kunne fileteres ned på køn, handicap, mm og sorteres både efter Tid og FINA points

# Teknisk:

## Backend og data:

Data vil blive gemt i Google Cloud ”Firestore” en NOSQL database som jeg ikke har nogle rigtige erfaringer med

Data arkitekturen vil derfor være ret forskellig for hvad jeg er vandt til og data skal tænkes lidt anderledes, jeg har allerede lavet et par idéer til data modeller se evt. Excel ark med data modeller.

Jeg er kommet frem til en model jeg tror vil fungere og hvor jeg vil kunne bruge til de scenarier jeg har sat op for hvad programmet skal kunne.

Dette skal jeg dog teste når jeg har fået noget data ind om det giver mening og om der er nogle huller jeg ikke har kunne gennemskue fra start, jeg skal dog med den opsætning være opmærksom på hvordan jeg læser dokumenter fra firestore så jeg ikke kommer til at få uhensigtsmæssig mange ”reads”

Så snart test af at det hele virker skal der sættes noget user accsses på således at alle ikke bare kan læse og skrive i min data 😊

## Frontend

Programmet skal lavet til et web UI og med tiden gerne en PWA (progressive web app)

### Angular:

Jeg vil benytte Angular som framework

Dvs. Det vil blive Scriptet i Typescript og alle ting skal laves i components / services

Her ligger en lang række udfordringer da jeg ikke kender Angular særlig godt og der er en del risiko i at jeg vil kunne opleve problemer jeg ikke vil vide hvordan jeg skal fikse.

Dataen frem og tilbage mellem Angular og Firestore er naturligvis også en udfordring

## Ui:

Angular material vil blive benyttet

Giver god mening at holde det i Angular material da de taler godt sammen og er lavet til formålet.

Ud over denne standard UI kan components styles individuelt om nødvendigt, jeg vil gerne have et meget clean minimalistisk og brugervenligt design så ”alle” kan finde ud af at bruge det.