

Indholdsfortegnelse

Sketches.....	2
Wireframes	2
Informations arkitektur og prototyping.....	3
Usability og design	3
<i>Enkelhed.....</i>	<i>3</i>
<i>Konsistens</i>	<i>3</i>
<i>Opdagelighed, struktur og klarhed</i>	<i>4</i>
<i>Ikoner</i>	<i>4</i>
<i>Farver.....</i>	<i>4</i>
Living Style Guide	5
Kode	6
<i>Grid</i>	<i>6</i>
<i>Funktioner.....</i>	<i>6</i>
Bilag.....	8
<i>Bilag 1</i>	<i>8</i>
<i>Bilag 2.....</i>	<i>8</i>
<i>Bilag 3.....</i>	<i>9</i>
<i>Bilag 4.....</i>	<i>9</i>
<i>Bilag 5.....</i>	<i>10</i>
<i>Bilag 6.....</i>	<i>11</i>
<i>Bilag 7.....</i>	<i>11</i>
<i>Bilag 8 (oprettelse af joke).....</i>	<i>12</i>
<i>Bilag 9 (visning af joke).....</i>	<i>13</i>

Sketches

Vi tegnede¹ alle mange idéer og sammenlignede med hinandens. Vi kom hele tiden den endelige idé lidt nærmere ved løbende ændringer. Vi snakkede om store og små funktioner. Til at begynde med havde vi en idé om, at brugerne skulle have mulighed for at udfordre hinanden i jokes, kommentere på jokes og mange andre ting. Men med henblik på Hicks Law, var det vigtigt for os at skabe et produkt, der er nemt for brugere i alle aldre at benytte. Vores udgangspunkt blev derfor, at vi ville lave få muligheder for hurtig forståelse. Derfor skrottede vi flere idéer og koncentrerede os om at lave det så simpelt som muligt. Det var aldrig meningen, at vi ville lave en kompleks app. Derfor begyndte vi at tænke kvalitet frem for kvantitet.

Med idéer om hvad vi syntes, vores app skulle kunne, skrev vi ønsker ned på papir. Alt fra sikre valg til idéer, vi med sikkerhed ikke kunne nå at kode og derfor have med i vores endelige produkt. En af vores oprindelige idéer er en funktion, hvor brugerne kan battle mod hinanden i jokes – 'jokesoff'. I begyndelsen var vi sikre på, at det var dét, vores projekt skulle ende ud i. Men længere henne i processen fandt vi hurtigt ud af, at det var for ambitiøst og ikke realistisk at ende med sådan en løsning. Vi gik derfor med udgangspunktet at lave en webapp, hvor det er muligt at tilføje jokes og se alle i et offentligt feed.

Vi søgte inspiration hos lignende apps² for at finde ud af, hvilke funktioner der oftest er lavet. Ud fra dette blev "min profil" en del af vores plan for, hvad der skulle indgå i projektet. Løbende tilføjede vi ekstra funktioner som notifikationer, søg og muligheden for at differentiere siderne i feeden.

Wireframes

Med udgangspunkt i vores endelige sketches og idéer, lavede vi wireframes. De skulle være vores udgangspunkt til både user experience og user interface. Til start lavede vi enkelte sider, og sideløbende med udarbejdelse af informationsarkitekturen, tilføjede vi flere og flere. Da vi havde vores ikoner færdige, blev disse implementeret.

Da vi havde lavet en skrabt udgave, begyndte vi at gå i dybden med den visuelle del. Det var vigtigt for os at skabe et meget overskueligt hierarki i vores valg af placering af knapper mm. Vi benyttede os af Law of Proximity³ (eller også Gestalts lov om nærhed⁴) samt søgte inspiration på andre platforme. Det var tydeligt at se, at menuer typisk er simple i opsætning og udtryk for nem forståelse. Knapper der hører til noget specifikt, er altid placeret nær det element, det hører til. Ud fra denne teori placerede vi vores ikoner og elementer.

Vi var også bevidste om andre designteorier. Bl.a. Millers Law og Serial Position Effect. Den første inddrog vi i vores valg ved, at vi valgte ikke at have mere end 7 +/- elementer i felt. Det ideelle er 5, og det valgte vi at gå med. Af denne årsag er der ikke flere vigtige funktioner i samme funktion som eksempelvis som bundmenu. Næste teori omhandler, at vi bedst husker første og sidste ting, vi ser. Derfor har vi i vores bundmenu valgt, at joke feed skal være den første funktion, og profil skal være den sidste. Vi valgte dog at gå en lille smule på kompromis med denne teori, da knappen 'post joke'

¹ Bilag 1

² Bilag 2

³ <https://medium.com/swlh/ux-laws-with-practical-examples-c418b4738d20>

⁴ <https://www.nielsgamborg.dk/?p=gestaltlovene>

er centreret. For at skabe opmærksomhed omkring denne, lavede vi den en lille smule større og sørgede for, at der er white space omkring, så den ikke bliver overset eller gjort mindre vigtig.

Informations arkitektur og prototyping

Med afsat i vores sitemap forsøgte vi at skabe et overblik over vores projekt via informations arkitektur. Vi skabte et kort⁵ for hvert step i vores app, så vi nemt kunne flytte rundt på funktionerne alt efter, hvad der gav mening. Vi fik medstuderende til at prøve at forstille sig, hvordan det vil se ud i et færdigt produkt, og om de forstod navigationen samt syntes, det var overskueligt og nemt. Vi fik et par rettelser til placering, så det kunne skabe en bedre forståelse.

Da overstående var færdigt, valgte vi at lave en prioriteringsliste⁶ over hvert step. Hvad vi ville være sikre på at have færdigt, og hvad der gav mest mening at have med, hvis vi ikke nåede i mål med det hele. Vi valgte at tage udgangspunkt i vores topmenu, der bl.a. indeholder joke feed. Det er det, vores produkt skal kunne. Næst vigtigt er bundmenuen. For at vores app kan fungere, skal der være en måde at levere indhold på. Derfor var post joke næste step i produktionen. De to funktioner er grundstenene, og med disse kunne vi have haft et produkt, selvom det ville være meget basic. Herefter kom alle andre muligheder som likes, top jokes, min profil, at kunne følge brugere, dele jokes, anmelde og meget mere. På denne måde har vi nemt kunnet overskue vores arbejdsgang og lave små sprints for hver ny funktion.

Usability og design

Helt fra starten af havde vi en idé om, hvordan vores færdige produkt skulle se ud. Helt bevidst besluttede vi os dog for ikke at lægge os fast på valg af design i begyndelsen, da det var vigtigere for os, at vi havde alle elementerne klar. Vi havde dog en del dialoger om, hvad vi ønskede.

Inden vi tog valg om design, gennemgik vi principper for webusability. Vi arbejdede ud fra en håndfuld af disse – bl.a. enkelhed, konsistens, opdagelighed, struktur og klarhed.

Enkelhed

Vores prioritering var at skabe en enkel interaktion. Produktet skulle være meget lige til, og der skulle ikke indgå overflødige funktioner og elementer, der kunne skabe forvirring for brugeren. Vores vision om idéen var, at man brug af appen første gang skulle kunne navigere rundt helt ubesværet. Af denne årsag undlod vi besværlige funktioner som fx ”jokeoff” (en konkurrence hvor brugerne kan battle mod hinanden i jokes).

Konsistens

Der findes så mange sociale apps i dag, og dem der er succesfulde, opfylder brugernes forventninger. Det var vigtigt at skabe et produkt, der var bygget op om det, vi allerede kender i forvejen. Derfor blev vi inspirerede af andres ikoner, opsætning, sprogvalg mm. På vores egen webapp benytter vi også samme elementer på de forskellige sider for at skabe ensartethed. Det øger forståelsen og skaber unødvendig tid på platformen.

⁵ Bilag 5

⁶ Bilag 6

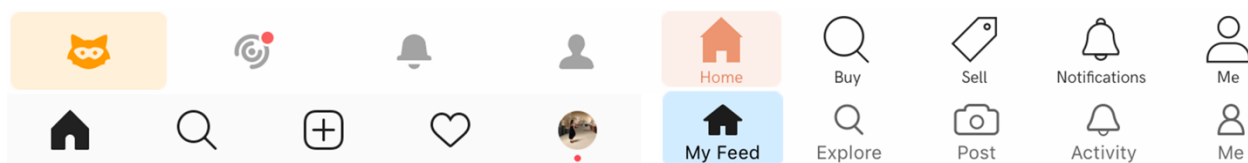
Opdagelighed, struktur og klarhed

Som tidligere nævnt brugte vi en del tid på at overveje, hvor ikoner, jokes og menuer skulle placeres. Disse funktioner er alle vigtige, og derfor skulle de have en synlighed og en velovervejet placering. Vi tænkte også meget over prominente elementer. Interaktive funktioner er markeret tydeligt for at vise, at de er vigtige og funktionsdygtige. Ligesom med Gestalt, loven om nærhed, ønskede vi en meget klar gruppering af elementer med tilhørsforhold. Det virkede som det mest logiske valg, og hvad brugerne nemmest kan navigere i. Både strukturen i indholdet er vigtigt, men også valg af komponenter har været nøje gennemtænkt, da vi ønskede, at disse skulle være selvforklarende og nemme at forstå.

Vi begyndte processen med at læse en del teorier om trends inden for design. Stort set alle artikler er enige om det meste. De ting der oftest bliver nævnt, er farver, typografi, logo og motion design. Vi tog derfor udgangspunkt i disse.

Ikoner

Vi valgte hvert især at bruge en dag på at søge inspiration til vores ikoner, der skulle være grundstenene i produktet. Vi havde alle eksempler med fra bl.a. Instagram, Twitter, Jodle og Flat Icons. Vi printede det hele ud for at få et bedre overblik. Det var vigtigt for os at tage udgangspunkt i Jacobs Law for at skabe lighed mellem eksisterende platforme med henblik på brugervenlighed og genkendelighed. Vi ønskede ikke at stræbe efter mere end originalitet.



Vi gik derfor med et meget enkelt design. Nogle af ikonerne tog vi online, og andre designede vi selv i Illustrator. Vi bemærkede, at stort set alle bundmenuer tydeligt viser, hvilken side brugeren befinder sig på for god brugervenlighed. Vi så en funktion på Instagram, vi ønskede at benytte. Når en bruger er på en funktion i bundmenuen, bliver ikonet markeret tydeligere for at vise, hvor i appen brugeren befinder sig. Vi tegnede derfor alle ikoner op med lidt tykkere streger for at skabe denne effekt. Vi forsøgte os også i forskellige design – bl.a. med +-knappen⁷.



Farver

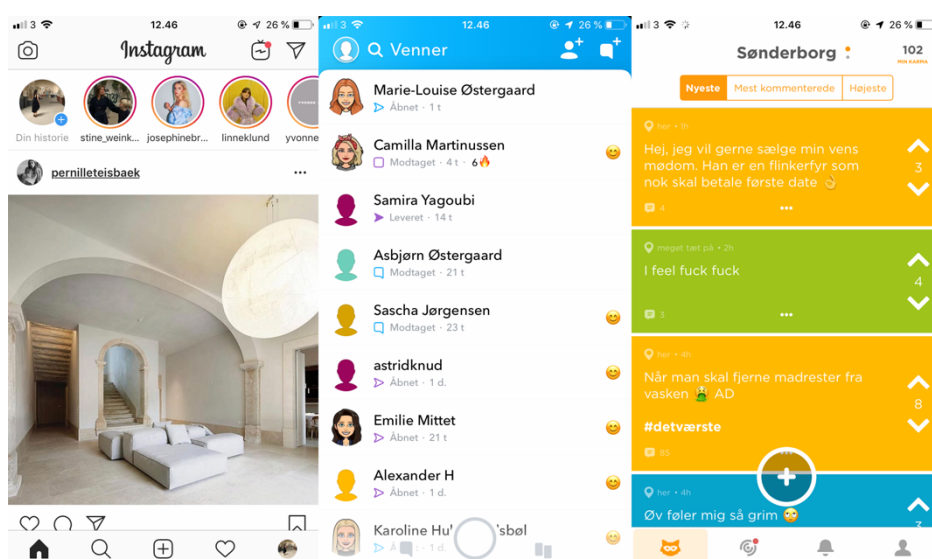
Den helt store trend for 2019 er farver. I UI ser vi meget stærke farver, der skaber opmærksomhed og larm. Med flere teorier om farver i mente, begyndte vi at udvælge via Adobe Color Wheel samt at søge inspiration online. Nogle af vores farver er taget fra en designvirksomhed, The Apartment, der er en førende designvirksomhed i København. Deres hjemmeside er fyldt med kontraster og spændende farver.

⁷ Bilag 3 og 4



*"Bright hues tend to be lighter less serious and "younger" than many other colors"*⁸

Ydermere kiggede vi på eksisterende apps, der henvender sig til samme brugere som vores. Vi så på bl.a. Jodle, Instagram, Twitter og Snapchat. De anvender alle klare, lyse farver. Instagram er gået mere væk fra at benytte det i høj grad, fordi de allerede har fået skabt sig et kæmpestort brand, og folk derfor kender dem.



I og med vi valgte at arbejde med lyse, stærke farver, var det vigtigt, at vi skabte plads omkring elementer og tekst, så det er overskueligt at kigge på. For hver farve vi har benyttet, har vi valgt en lidt mørkere tone til at understrege vigtige funktioner. Fx har vi valgt, at vores header og footer i hver joke-boks skal være mørkere for at vise, at de har interaktiv funktion. Dette har vi også valgt at benytte i vores trykbare menuer. Topmenuen har et aktivt link. Når den er valgt og er i brug, vises dette med en mørkere kontrast ift. normalfarven. I vores bundmenu har vi valgt en mere simpel løsning. Den skal skille sig ud, fordi den har nogle vigtige funktioner og navigationsmuligheder. Den har derfor en hvid baggrund og sorte ikoner. Når brugeren trykker på fx feed, vil ikonet blive markeret med fed. På denne måde skabes der altid et overblik over, hvor man er.

Living Style Guide

Vi havde flere valgmuligheder for at dokumentere vores kode, men beslutningen om at lave en Living Style Guide var nem. For det første gav det os en struktureret og ny måde at tænke på, i og med vi startede med at designe og kode vores komponenter, inden vi satte dem sammen. Ydermere var det

⁸ <https://uxplanet.org/2019-ui-and-ux-design-trends-92dfa8323225>

overskueligt at have et sted, hvor vi har alt samlet og hurtigt kan vende tilbage til, hvis vi mangler noget specifikt eller skal skabe et overblik.

Inden vi begyndte at kode, startede vi med at opsætte et grid til vores guide. Efterfølgende lavede vi kategorier og gav dem passende classes. Da vi havde strukturen på plads, tog vi udgangspunkt i vores prioriteringsliste. På baggrund af denne startede vi alle med at kode de forskellige komponenter som for eksempel bundmenuen, knapper, topmenu til forsiden/feeder, kasserne der indeholder jokes, topmenuen i min profil og så videre. Da alt var færdigt, begyndte vi at lave ændringer i de visuelle design.

Kode

Vi aftalte at skrive kommentarer i vores kode hver især for et lettere overblik, når vi skulle hjælpe hinanden og til sidst sætte vores dele sammen i samme fil. Det har gjort arbejdsgangen meget nemmere.

Webappen blev opbygget ud fra komponenter, så vi til hver en tid kunne tage noget ud, eller tilføje noget til webappen. Det gør, at vi har genbrugt meget kode, både i html og css. Her har vi gjort lidt brug af BEM, hvor vi dog ikke har nogle særlige modificers, da vi bare har sat stylingen af block komponentet sammen med elementet. Her har vi tænkt meget atomic design, men dog mere i molekyler og organismer – ikke så meget i atomer. Det gør også at meget af vores css har kunne blive samlet og gøre brug af classes. Sammen med brugen af BEM, er alt vores indhold sat ind i en class, undtagen vores body og ét enkelt em tag. Vi har tænkt over semantik, da vi lave markup af html, så gav de tags, som vi synes gav mest mening i forhold til det indhold det havde og de funktioner, som der skulle være. Dog havde vi ét sted, i vores en af vores genbrugte komponenter, hvor vi ikke vidste, hvad det ville give mest semantisk mening. Derfor er det nu her en div container – som vi er vel klar over ikke er en optimal løsning og nok kunne have været alt andet.

Grid

På vores body har vi opsat et grid⁹ for strukturen af alt indholdet. Her har vi valgt at gå med 12 kolonner, som hver har 1fr. Det har gjort det nemmere, at placere indhold, som var lidt anderledes end resten og funktioner som vi nødvendigvis ikke fik implementeret. Med et overordnet grid hjælper det med det responsive af webappen, når man går fra de forskellige mobil størrelser. Dog med det indhold og funktioner, som vi endte med, kunne man nøjes med nogle færre kolonner i dette grid, da vores komponenter har en kolonne span på 12.

Det har været lidt et problem med rows, da vi tit endte med at have meget whitespace imellem de forskellige komponenter. Vi ville ikke definere konkrete rows, da nogle så kunne være et fast antal pixels, imens andre skulle have en auto højde, i forhold til hvad for en side man var på. Inde i de forskellige komponenter, har vi brugt nogle nested grid, for at gøre det nemmere for os, at placere de forskellige elementer. Også så det skaleret rigtigt på andre skærmstørrelser.

Funktioner

Den vigtigste funktion ved webappen, at er kunne post en joke. Det var også en af de funktioner, hvor vi blev nød til at bruge andres kode, som vi modificeret til vores behov. Her blev vi nød til at gøre brug af sessionStorage, til at gemme den indtastet joke på refresh. Det var kode, som var lidt

⁹ Bilag 7

skræddersyet til vores behov, så det var lige til at implementer og modificer, så det passede. Dog fik vi lidt senere problem med, at skulle vise den gemte joke på en anden side. Derfor endte vi med at bryde noget af koden op, så vi nu har en separat scriptfil til at oprette joke, hvor koden til at vise den, er inde i vores primær script fil (se bilag). Ellers har vi mange click events til forskellige funktioner. Her har i stødt ind i lidt udfordring senere hen, da de forskellige komponenter blev lavet, med tilhørende javascript¹⁰¹¹, var der ikke taget højde for, at det ikke kun skulle virke på et element. Så vi måtte modificere dem lidt og sætte dem ind i nogle loops. Nogle funktioner fik vi hellere ikke implementeret, men som vi havde snakket om eller gerne ville have haft med. Her har vi en splash screen. Den valgte vi at slå fra, da vi ikke havde tid til sidst at få det til, kun at komme én gang – selv efter refresh.

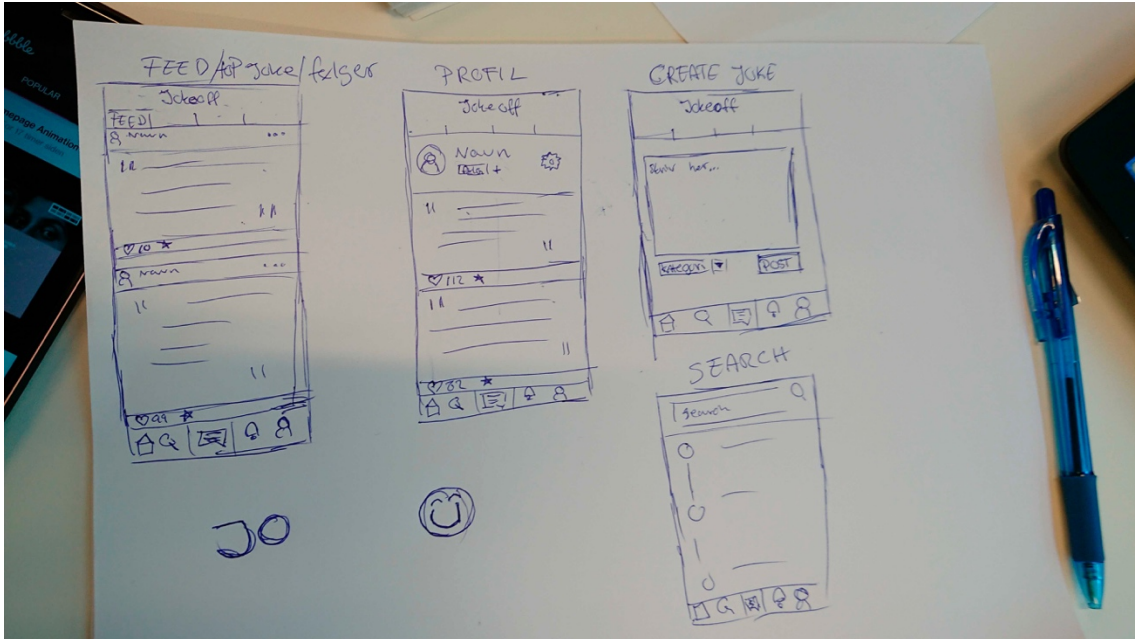
Vi ville have haft store versioner af jokesen, når man trykkede på dem i feedet. Topmenuen er heller ikke helt funktionel, da indholdsmæssigt ville være det samme som selve feedet, ud over at der ville have været noget sortering. Vores like counter virker ikke optimalt, da vi i sidste ende ikke havde tid til at få den til at virke i et loop og med den random like number, som vi nu har.

¹⁰ Bilag 8

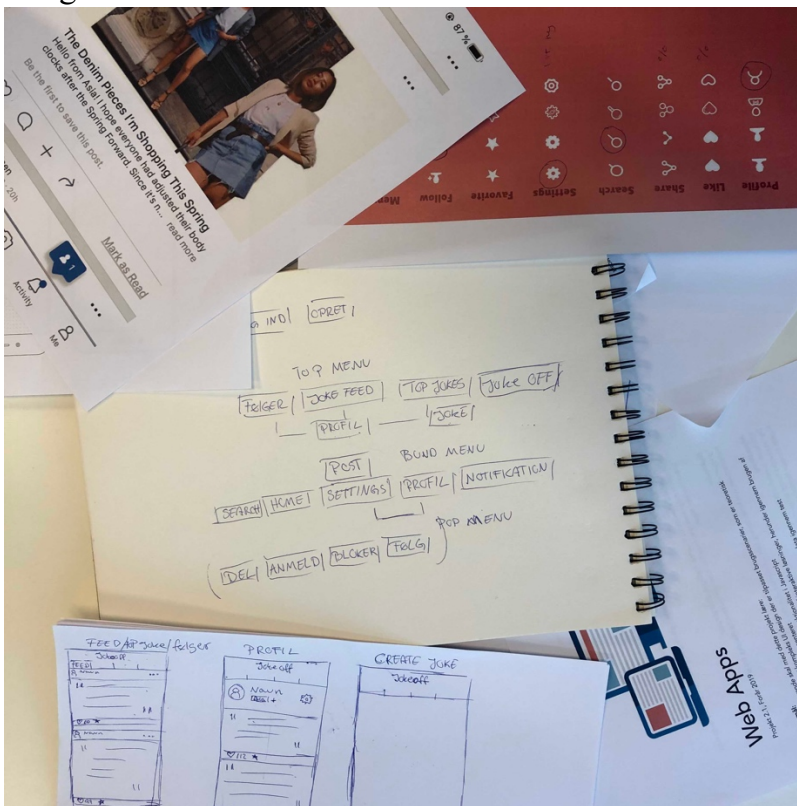
¹¹ Bilag 9

Bilag

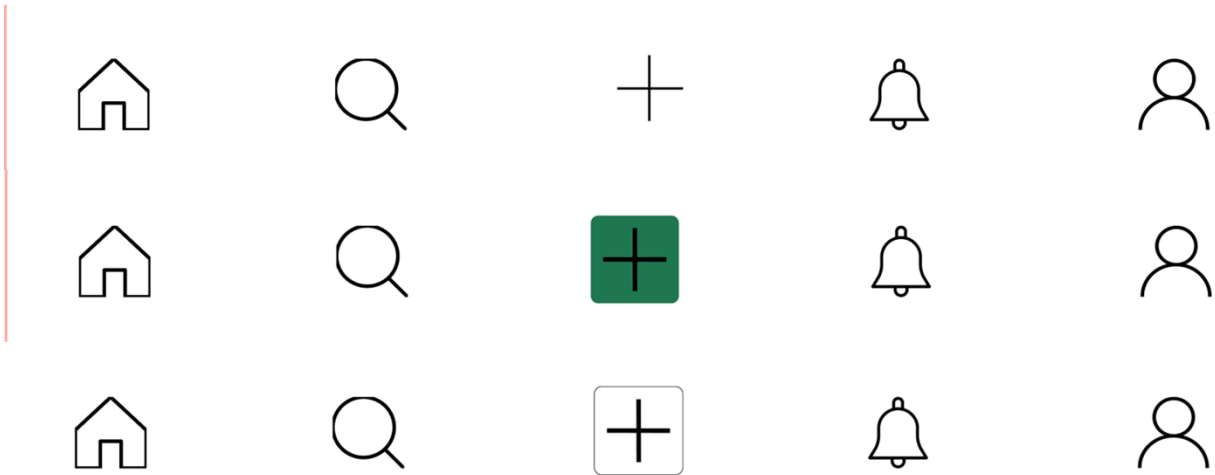
Bilag 1



Bilag 2



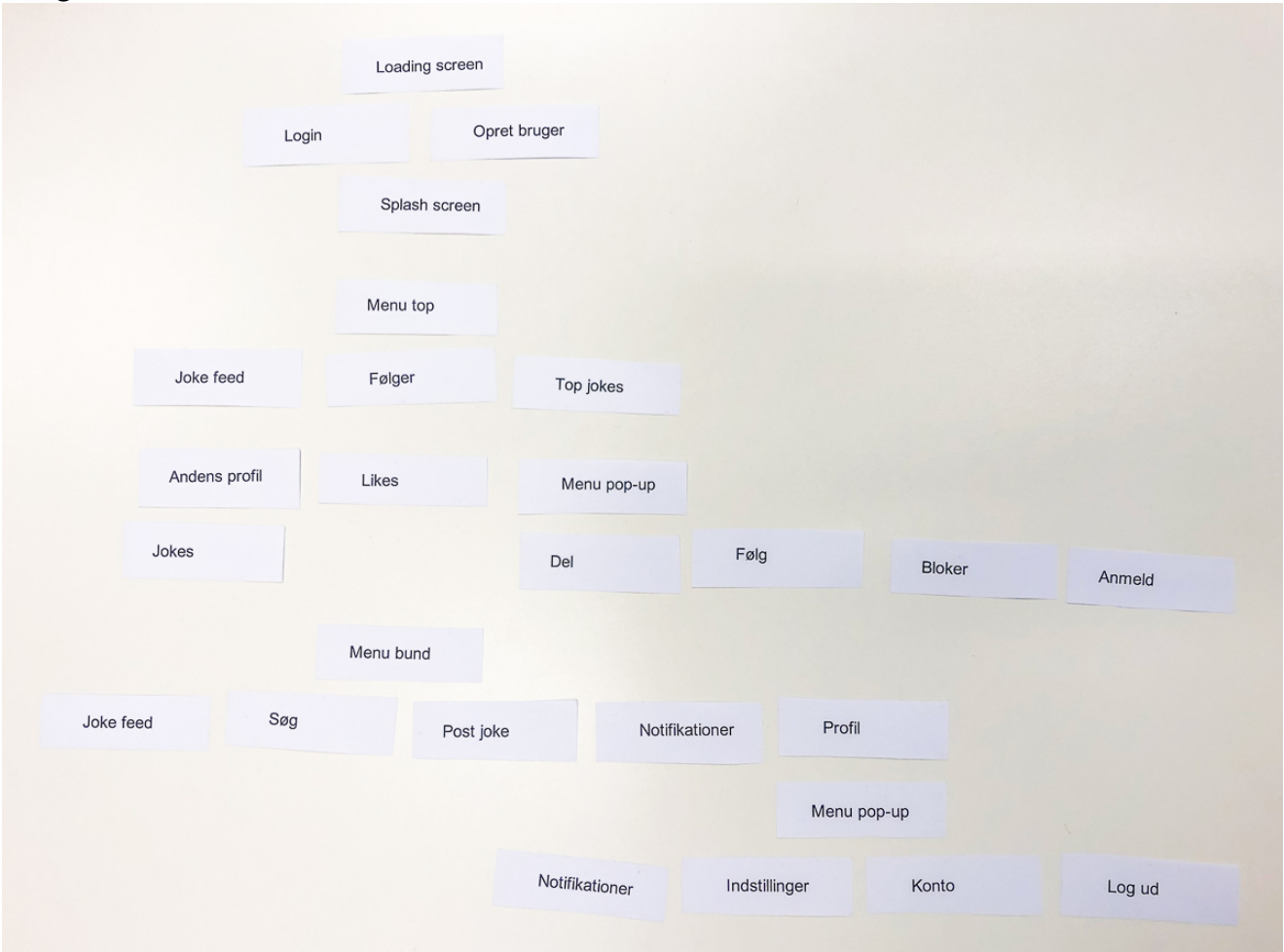
Bilag 3



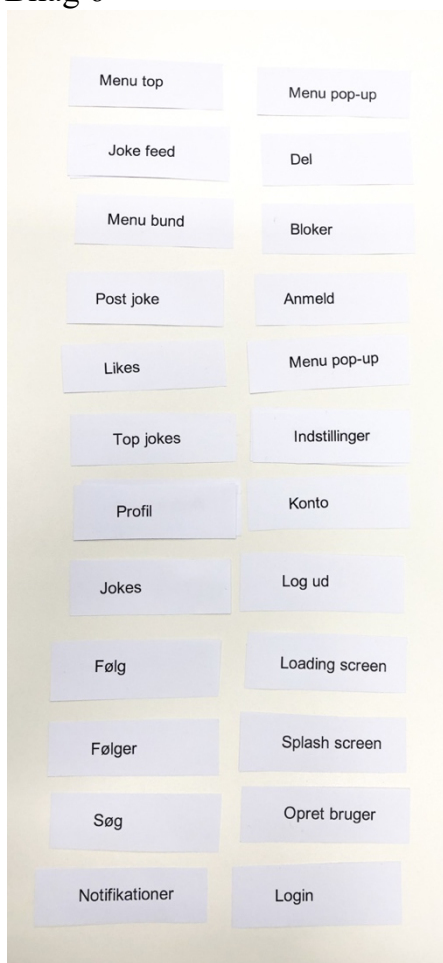
Bilag 4

JOKES	TOP	FØLGER
Jokes	Top	Følger
Jokes	Top	Følger

Bilag 5



Bilag 6



Bilag 7

```
body{  
  display: grid;  
  grid-template-columns: repeat(12, 1fr);  
}
```

Bilag 8 (oprettelse af joke)

```
let jokeSubmit = document.querySelector(".jokeInput");
let jokeInput = document.querySelector(".writeJoke");
let showJoke = document.querySelector(".showJoke");

// tomt array til at gemme jokes
let storedJokes = [];
// if jokes are stored in session storage, load them from there...
if(sessionStorage.getItem("jokes")) {
    storedJokes = JSON.parse(sessionStorage.getItem("jokes"));
}

// handle submit
jokeSubmit.addEventListener("click", function (e) {
    // create new joke object
    console.log("clicked");
    let tempObject = {
        joke: jokeInput.value,
    };
    console.log(tempObject);

    // add to array of jokes
    storedJokes.push(tempObject);
    console.log(storedJokes);

    // save to session storage
    sessionStorage.setItem("jokes", JSON.stringify(storedJokes));
});
```

Bilag 9 (visning af joke)

```
let jokeSubmit = document.querySelector(".jokeInput");
let jokeInput = document.querySelector(".writeJoke");
let showJoke = document.querySelector(".showJoke");

let storedJokes = [];  

// if jokes are stored in session storage, load them from there...
if(sessionStorage.getItem("jokes")) {  
    storedJokes = JSON.parse(sessionStorage.getItem("jokes"));  
}  
console.log(storedJokes);  
    // Display the jokes  
    for(let i=0;i<storedJokes.length;i++) {  
        // create a new article tag  
        let article = document.createElement("article");  
        article.classList.add("jokeContainer");  

        // Set the content of the article  
        article.innerHTML =  
        '<a class="jokeContainer__header">' +  
        '+' +  
        '+<h3 class="jokeContainer__username">JokeMaster69</h3>' +  
        '+' +  
        '+</a>' +  
        '+<pre class="jokeContainer__joke">' +  
        + storedJokes[i].joke +  
        '+</pre>' +  
        '+<div class="jokeContainer__footer">' +  
        '+' +  
        '+<p class="likesNumber"></p>' +  
        '+' +  
        '+</div>' +  
        ;  
        //console.log(article);  
        showJoke.appendChild(article);  
    }  
}
```