Med de basale idéer om hvad systemet skulle kunne, begyndte processen for at gå fra et upræcist koncept, til et produkt med pålidelige funktioner der opfyldte virksomhedens krav, og mere til. En proces der kan beskrives som en peberkværn, der kværner de ukonkrete koncepter og idéer til fine praktiske løsninger, der modsat koncepter og idéer kan løse virkelige problemer, når den drejes omgang efter omgang. (<- shit det corny alligevel, ved sgu ik)

Ved hjælp af daglige standups, research af vores værktøjer og en hel del trial and error på programmeringsfronten, arbejdede vi os stille og roligt imod noget mere og mere konkret. I starten af denne fase gav vi os i kast med at designe det første udkast til en brugbar database, med ER-diagrammet som et godt værktøj til at visualisere, og løbende tilpasse vores idéer under denne proces.

Imens arbejdet på databasen var undervejs, blev der i den anden halvdel af gruppen arbejdet på den visuelle del af programmet, både ved at forsøge at videreudvikle vores skitser til mere finpudsede prototyper af de forskellige dele af systemets brugergrænseflade, men også ved det rent visuelle design i form af et løbende opbygget CSS sheet. Denne del fyldte meget i denne indledende fase af konstruktionen, men da de mest gængse elementer som knapper, tekstfelter og de grundlæggende farver var på plads, var denne rent visuelle del af designet noget der sporadisk blev opdateret, i forbindelse med vi tilføjede flere og flere elementer til brugergrænsefladen. Farverne blev udvalgt efter virksomhedens eget ønske, og flere andre designelementer var løst inspireret af deres egen hjemmeside. Dette skulle medvirke til at brugerne kunne få en oplevelse af at tingene hænger sammen.

Med hensyn til de konkrete krav til funktionaliteten fra virksomhedens side, blev der igennem denne fase, som nævnt tidligere, flere gange ændret i designet. For eksempel var det i starten planen at systemet i forbindelse med opgaverne, skulle gemme en opgavetekst direkte i programmet, samt videoer og billeder. Dog blev dette en stor problemstilling, med hensyn til visuelt at opsætte opgaven, så den fremstod som virksomheden måtte ønske. Spørgsmål vi diskuterede i mellem os på daværende tidspunkt, lød på hvordan vi skulle tilpasse opsætningen, hvis en opgave måtte have et vilkårligt antal både videoer og billeder, samt om det var for begrænset kun at tillade ét af hvert af disse elementer på hver opgave. Sidstnævnte ville gøre systemet særdeles rigidt, og det var vi ikke interesserede i. Vi fortsatte derfor vores søgen, og et af vores store gennembrud i denne designfase var, da vi fandt ud af vi kunne opbevare filer i databasen.

I den næste fase af systemets overordnede design, fastslog vi på baggrund af vores nyerhvervede viden om opbevaring af filer såsom word-dokumenter, at det ville være lettere for virksomheden at opbevare disse direkte i programmet. Vi vidste fra et møde med dem, at de allerede på daværende tidspunkt opbevarede og udsendte deres opgaver i dette filformat, og det virkede oplagt at arbejde videre på den information. Udover fordelen ved at virksomheden let kunne bruge deres allerede etablerede opgavebibliotek, havde det også den fordel, at de lettere kunne konstruere tekst og billeder som de havde behov for, indenfor word-dokumentets velkendte rammer. Hernæst var der tilbage at finde en måde at tilknytte videoer til opgaverne, da dette også var et krav fra virksomhedens side.

Det overordnede design af opgaverne var nu på plads. Et fleksibelt materialebibliotek bestående af videoer og word-dokumenter, kunne let tilpasses af virksomheden som ønsket, ved at tilføje de ønskede materialer til databasen, hvorefter de kunne tilføjes til opgaver. Herefter vendte vi blikket imod et andet af virksomhedens centrale krav, nemlig muligheden for at kategorisere opgaverne ved hjælp af tags.­­­­­

I vores løbende sammenspil under denne proces, kom vi frem til at det var oplagt ikke kun at give opgaverne tags, men også at lade materialerne have del i samme tags. Formålet med dette var at tilføje en ekstra dimension til virksomhedens muligheder for at organisere deres materialebibliotek, da man let kunne forestille sig hvordan for eksempel én video, ville kunne gøre sig gældende på flere variationer af en lignende opgave. Her gav det god mening at have materialerne adskilt fra opgaverne, men samtidig at have muligheden for let at finde videoen i materialebiblioteket, når den skulle tilføjes til en ny eller eksisterende opgave. Igen prioriterede vi systemets fleksibilitet højt.

Efter den mere overordnede del af beslutningerne om hvordan systemet skulle leve op til virksomhedens krav, gik vi naturligvis ind i en fase af mere praktisk arbejde med at udvikle systemet. Her gjorde vi os tanker om hvordan loginsystemet skulle fungere i praksis, og om hvorvidt vi skulle prioritere en vis form for sikkerhed. Med netop sikkerhed for øje, brugte dele af gruppen nogle dage på at undersøge koncepter som hashing og salting, for at finde ud af om det var noget vi skulle implementere i forbindelse med kryptering af passwords. Vi kom senere frem til, at dette ikke var det vigtigste for systemet på daværende tidspunkt, og vi valgte at bruge vores tid på andre dele af programmet, endnu engang med målet om at opfylde flere af virksomhedens krav i tankerne.

Omkring denne fase brugte vi en hel del tid på at sørge for at vi befandt os på samme side, når det kom til vores forståelse af hvordan systemet skulle se ud. En af tingene vi diskuterede heromkring, var hvorvidt det skulle være muligt for én bruger i systemet, at have flere roller. På trods af at vi gerne ville implementere dette, besluttede vi at vi kunne arbejde mere på programmets konkrete funktionalitet, materiale og opgaver, og manipulation af disse, ved at gå på kompromis med det.

Efter vi fik alle disse områder banket på plads, kunne vi se et mere konkret produkt forude, og for alvor påbegynde arbejdet på den rene programmeringsdel af systemet. Selve programmet var opbygget af et datalag og et præsentationslag. I præsentationslaget husede der både fxml-filer, controllerklasser, samt et CSS style sheet. Alt kommunikation imellem brugergrænsefladen og databasen, foregik igennem datalaget – dette sidstnævnte tog form som én klasse, med metoder der ved hjælp instanser af klassen kunne bruges i resten af programmet. Ved siden af disse, består programmet også af en række objekter, der har til formål at hjælpe os med at transportere information rundt i programmet. Her var det oplagt for os at lave en abstrakt klasse ”object”, der kunne nedarve til en stor del af vores andre klasser, der alle delte både et id som primærnøgle i en tabel, og en titel.

I grove træk, bestod arbejdet herefter af en masse benarbejde med programmering af alle de ønskede funktioner. En stor del af dette bestod i at vise og manipulere objekter, igennem lister. Et andet interessant problem, der krævende en del undersøgelse, var implementeringen af upload og download af filer til systemet.