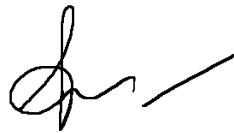


DOG GOOD

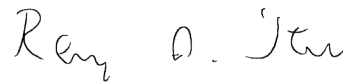


Vejleder: Tuje Becher Gruppe nummer: 07 Tegnsætning: 31544

Anastasia Keller
cph-ak186@cphbusiness.dk



René Abel Jensen
cph-rj135@cphbusiness.dk



Augusta Lindskov Naundrup-Jensen
cph-an126@cphbusiness.dk



Benjamin Lipsø
cph-bl91@cphbusiness.dk



Sabine Larsen
cph-sl176@cphbusiness.dk



Indholdsfortegnelse

1. Problemformulering
2. Projektplanlægning (SCRUM)
 - 2.1 SCRUM
 - 2.2 Burn-down chart
3. Konzeptudvikling
 - 3.1 Brand identity - brandnavn
 - 3.2 Inspiration - moodboard
4. Design brief
5. Informationsarkitektur
 - 5.1 Website navigation diagram
 - 5.2 Sketches, wireframes og mockups
6. Design process
 - 6.1 Logodesign
 - 6.2 Design Manual
7. Crowdfunding projekt
8. Userstories
9. Database opbygning
 - 9.1 ER-diagrammer
 - 9.2 ER-diagram på 3.NF
 - 9.3 Data til database
 - 9.4 Attribut Tabel
 - 9.5 Sql views til queries på hjemmesiden
 - 9.5.1 Forside
 - 9.5.2 Projekter
 - 9.5.3 Kategorier
 - 9.5.4 Ranking
10. Interaktionsudvikling (kodning)
 - 10.1 Kode struktur
 - 10.2 PHP
 - 10.3 Login system
11. Konklusion og refleksion
12. Referencer
13. Bilag
14. Gruppekontrakt (muligvis et bilag)

1. Problemformulering

Formålet med projektet er at skabe en prototype til en crowdfunding hjemmeside. Prototypen skal fungere med database, og give brugere mulighed for at se, støtte og oprette projekter via siden.

2. Projectplanlægning

2.1 SCRUM

SCRUM er en planlægningsmetode der bliver brugt til at få overblik over hvor lang tid man har til at færdigøre hver opgave. Hver opgave får hver sit punkt hvor derefter 3 personer skal udgøre hvor lang tid de mener denne opgave til komme til at tage. Man lister derefter talene op i arket, starter med optimisten og slutter med pesimesten. Arket udregner derefter hvor lang tid man ca har til opgaven hver i sær.

Vi startede med at lave vores SCRUM som noget af det første, hvor vi forsøgte at sætte alle opgaverne op og angive hvor meget tid vi mente at vi skulle bruge på hvert punkt.

Som projektet skrider hen har vi måtte sætte flere punkter på som inden vi havde overset eller punkter vi senere mente var nødvendigt for opgaven. Se bilag 01

2.2 Burn-down Chart

En burn-down chart er en graf som ud fra informationerne fra SCRUM viser hvordan arbejdsprocessen har været.

Vores burn-down chart viser at vi i starten var en lille smule bagud i starten af projektet, men har efterfølgende været med på den tidsplan der er blevet sat.

Man vil også lægge mærke til at det sidste punkt rammer under 0-punktet.

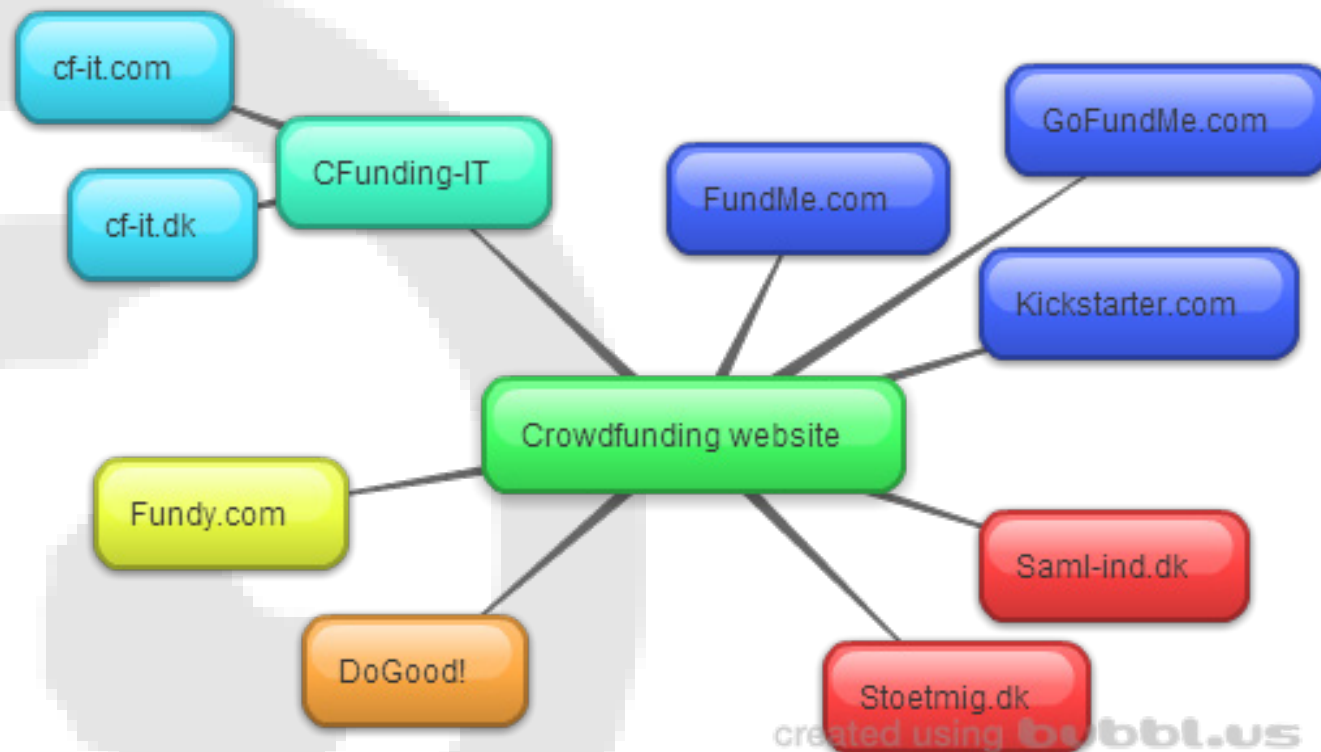
Dette er et resultat af at vi har brugt mere tid end hvad vi oprigtigt mente var nødvendigt. Se bilag 02

3. Konceptudvikling

3.1 Brainstorm omkring et brand navn

Vi har besluttet at vælge et brandnavn til vores hjemmeside for at gøre det lettere at udvikle og designe den visuelle identitet. Som vi har forstået fra opgaven var det ikke meningen at man bruge navnet på klient-firmaet CFunding-IT til et crowdfunding template-website. Dette lyder ikke så interessant som navnevalg.

Vi har lavet brainstorm omkring valget af et brand navn, som vi har dokumenteret i en Brainstorm-chart, der kan ses nedenunder.*1



Ideerne i chartet er fremhævet med forskellige farver og der er grupper af forskellige kategorier. Fx er navne som vi fandt uinteressante markeret med lyseblå. Navne på hjemmesider der står i blå, findes allerede. Rød farve viser navne ideer på dansk - selv om har vi valgt dansk som hjemmesidens sprog, har vi i tankerne at hjemmesidens målgruppe kunne udvides yderligere til et internationalt marked. Derfor giver det ikke mening at navnet på hjemmesiden er dansk.

3.2 Valg af brand navn

Vi havde sorteret os frem til 2 muligheder til sidst: Fundy og DoGood. Vi kunne meget godt lige Fundy først, men fandt ud at ordet selv betyder "en kristen fundamentalist"*2, hvilket ikke ville passe til vores ide af hjemmesiden. Desuden findes der allerede en crowdfunding hjemmeside fundly.com, og det ligger sig meget tæt op ad "fundy".

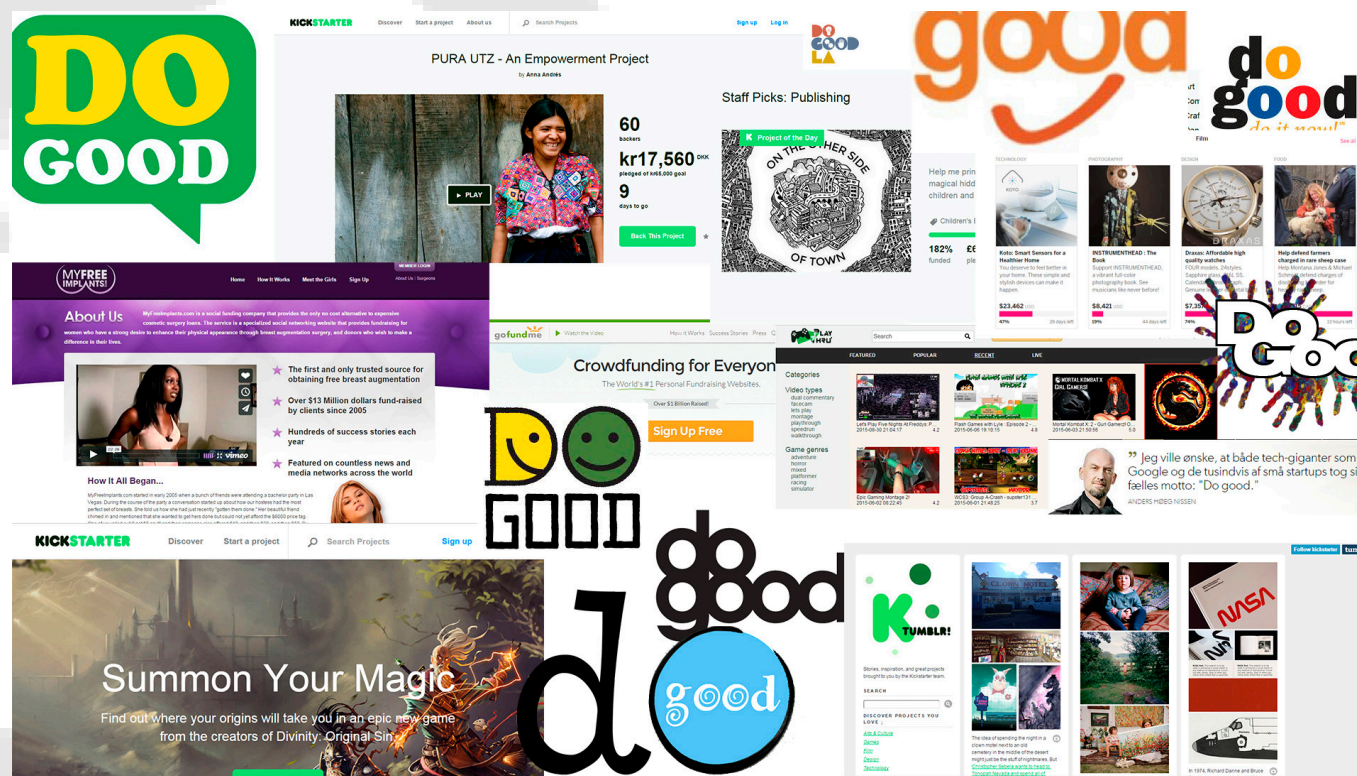
Derfor stod DoGood som enestående mulighed til sidst. Der var god grund til at vælge det som navnet til vores hjemmeside. For det første lyder det behagligt og enkelt. DoGood oversættes på dansk til "gøre noget godt". Der er flere betydninger i Googles ordbog:*3

"do good"

1. Act virtuously, especially by helping others.
2. Make a helpful contribution to a situation.
3. " Could the discussion do any good?"

3.3 Moodboard

Vi har taget inspiration til brand navnet og andre ideer og elementer på hjemmesiden fra forskellige billeder og hjemmesider (for det meste andre crowdfunding hjemmesider) på nettet. Vi har samlet en del af dem i et moodboard, som kan ses herunder.



4. Design Brief

Efter vi var færdige med problemformulering, konceptudvikling og definition af brugerkrav, ville vi opsamle hovedpunkter omkring projektets mål, krav og andre grundlæggende specifikationer i et dokument - Design Brief. I Design Brief besvares Laswells 5 spørgsmål: Hvem siger Hvad til Hvem gennem Hvilket medie og til Hvilken effekt? Disse spørgsmål er basale for den klassiske Laswells kommunikationsmodel. *4

Vi har udvidet de 5 spørgsmål med et par flere for at skabe bedre forudsætninger til projektets design trin.

Hvem er klienten?

CFunding-IT - et lille mediebyrå beliggende i indre København.

Hjemmesidens brand navn: DoGood

Hvad er projektets mål?

At udvikle website-prototype til crowdfunding system:

- backfront - database
- forefront - hjemmeside

Hvem er vores målgruppe?

Primær: projekt-oprettere og støttere (eller backers), samt andre fællesskabsorienterede mennesker, samt folk interesseret i innovation.

Backers-profil: danskere som primært ligger inden for den grønne sektion i Minerva-modellen. De er fællesskabsorienterede, vil støtte en god sag og hjælpe gode initiativer. De er nysgerrige og dynamiske. Mange af dem er projektstarternes venner og familiemedlemmer.

Projekt-opretteres profil: danskere som ligger inden for den grønne og blå

sektion af Minerva-modellen. Iværksættere, dynamiske achievers, også fællesskabsorienterede (især i velgørenhedsprojekter).

Begge segmenter skal have penge til rådighed (lidt overskud), tekniske muligheder (computer eller lignende med internetadgang) og være aktive brugere af nettet (mellem 18– 50 år). Livsstil: interesseret i nye ting, aktiviteter, produkter og generelt nye muligheder.

Sekundær: ældre generation som ville støtte, men er ikke så aktive på nettet (pensionister - 50+) alle andre og internationale folk i fremtiden.

Hvad er brugers oplevede værdi?

Backers:

- Jeg hjælper andre
- Jeg støtter en god sag
- Jeg støtter op om fællesskabet og gør livskvaliteten bedre
- Det er spændende at støtte nye ideer/teknologi (opfylder sin nysgerrighed)

Projektstartere:

- Jeg hjælper andre med mit projekt
- Det er spændende at starte noget nyt
- Det er tilfredsstillende at nå mit funding-mål med at samle penge til et projekt
- Hvis folk donerer penge nok, betyder det at min ide er god nok og kan udvikles videre

Hvad er budskaber til den primære målgruppe?

Til backers: "Støt dit fællesskab ved at støtte spændende projekter som gør livskvaliteten bedre!"

Til projektoprettere: "Vi hjælper folk med at indsamle penge til deres spændende, hjælpsomme og innovative ideer/projekter, som gør livskvaliteten bedre! Gør dine ideer til virkelighed!"

Gennem hvilket medie? Hvad er produktet?

Hjemmeside på dansk (på engelsk i fremtiden) med oversigt over forskellige projekter og mulighed for at registrere brugere, projekter samt at støtte projekter. (Se mere i brugerdefinerede krav senere)

Til hvilken effekt?

Brugerne skal oprette flere projekter, som kan få flere funds. Alle registrerings- og posting systemmer skal være velfungerende og effektivt. Det skal være let at finde projekter, evt. ved hjælp af kategorier.

Brugerdefinerede krav til hjemmesiden:

Som backer kan jeg:

1. Oprette bruger og logge ind
2. Opdatere mine brugeroplysninger
3. Se projektbeskrivelse
4. Betale ved at trykke på en støt-knap og via betalingsformular
5. Se betalinger/projektstatistik (i forhold til funding-mål)
6. Gå til projekternes sider på andre sociale medier
7. Bruge Search-funktionen for at søge ved projektkategorier, projektnavn, etc..
8. Læse nyheder/opdateringer på projekter
9. Kommentere på projektsider Som organisation/projektopretter kan jeg
10. Se andres kommentarer på projektsider

Som organisation/projektopretter kan jeg:

1. Oprette bruger / logge ind
2. Uploade og opdatere oplysninger om projekter
3. Uploade billeder/video til egne projektsider
4. Se betalinger/projektstatistik (i forhold til funding-mål)
5. Modtage betalinger fra sponsorer/backers
6. Se oversigt over backers/sponsorer som har betalt til et projekt
7. Se betalingers status (bekræftet eller ej)
8. Bruge søgefunktionen til at søge ved projektkategorier, projektnavn, etc
9. Læse nyheder/opdateringer på andre projektsider
10. Opdatere nyheder om egne projekter på projektside
11. Kommentere på projektside
12. Se sponsorerers/backeres kommentarer på projektside

Som admin kan jeg:

1. Se oversigten over organisationer og deres projekter
2. Se oversigten over brugernes oplysninger og betalinger
3. Se betalingsstatus (om betalinger var godkendt hos banker - bruges til at vise statistik på projekt-siderne)
4. Uploade, opdatere, redigere og slette nyheder/opdateringer om projekter (skriv når mål er opnået eller lignende)
5. Administrere kommentarfelt (CRUD)
6. Tjekke at indholdet på projektsider er censureret (fjerne upassende videoer, billeder eller tekst)
7. Stå for sikkerhed på brugeres profiler (passwords og betalinger)

Krav til design:

Stil: flad grafik, dynamisk, moderne, noget nyt, nysgerrig, simpelt, optimistisk

Farver: Orange og sort/grå. Lys (hvid) baggrund på hjemmesiden. Orange står for nyt, iværksætterånd, moderne, dynamisk og optimistisk. Andre farver er udelukkende til kontrast.

Typografi: logo skrifttype: moderne, letlæselig, original. Andre skrifttyper vælges baseret på logoets skrifttype.

Visuelle elementer: Logo, funktionelle ikoner, billeder som præsenterer projekter (mindst 1 projekt), knapper (Støt knap, Opret bruger-knap, log-in, etc.), statistikbar eller ikon

Tekster: på dansk, hovedbudskab på forsiden. Tekstrige sider: Om os, projektside
Formularer: Opret bruger, log-in, betaling, kontakt

* Mere detaljeret funktionelle og non.funtionelle krav FURPS+ kan ses i Bilag 3. *5



5. Informations Arkitektur

I denne sektion beskriver vi udarbejdelse af hjemmesidens struktur. Det handler om hvor mange sider der skal være, hvordan de hænger sammen og hvordan de skal se ud. Vi har sat prioriteter på siderne, som en del af Risk Management, hvor vi dropper de mindst vigtige sider, hvis vi kommer til at mangle tid. Prioritering ses herunder:

Sider på hjemmesiden:

Første prioritet:

- Forsiden
- Alle projekter (med kategorimenu)
- Projektsider
- Profilside
- Login-formular
- Opret bruger formular (registrer)
- Projektrettelses formular

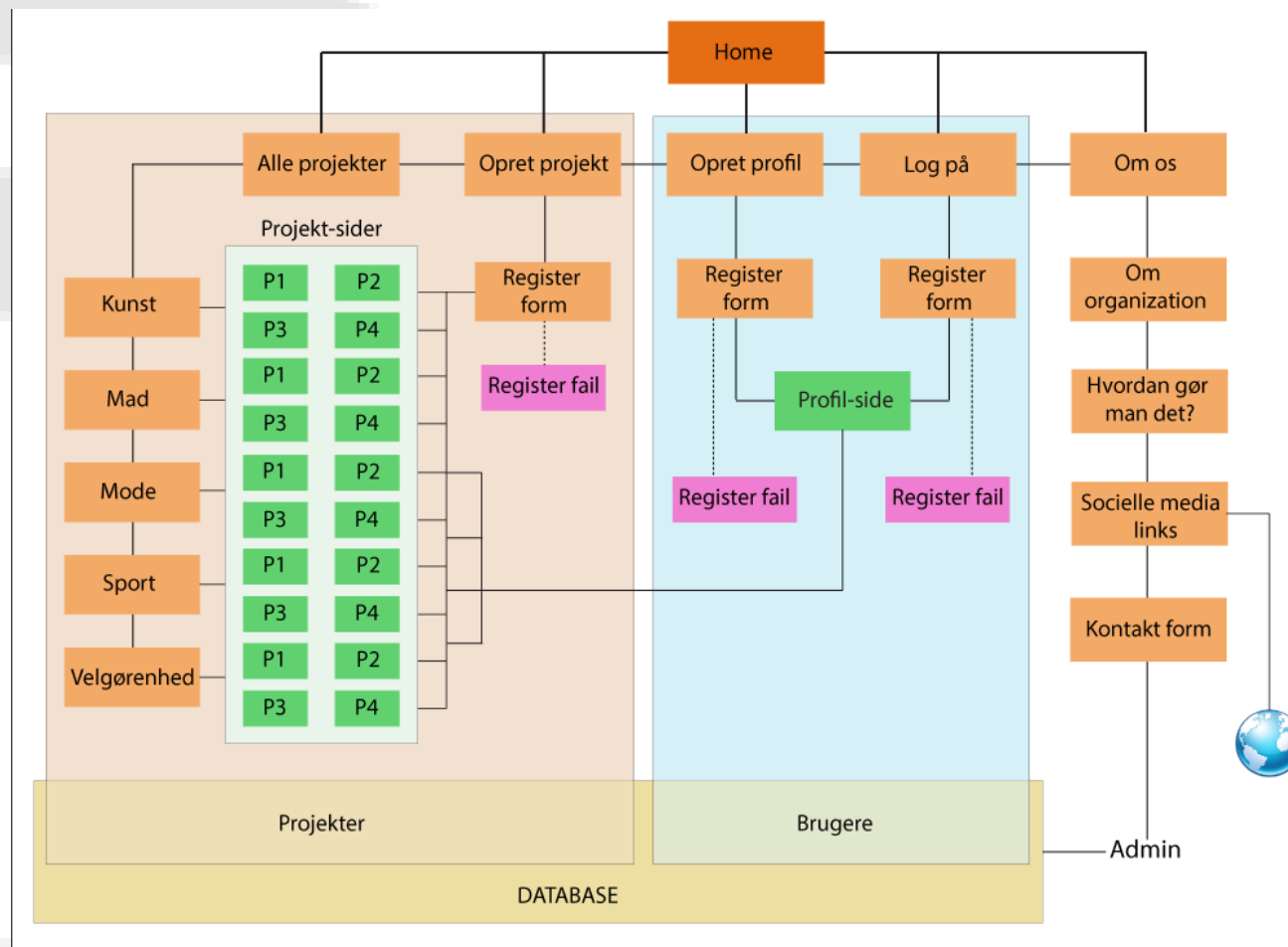
Anden prioritet:

- Betalingsformular
- Om os
- Kontaktformular

5.1 Website navigation diagram

Vi har lavet et website navigation diagram for at vise hjemmesidens struktur. Den viste sig at være mere kompliceret end først antaget, da hjemmesiden inkluderer flere registreringsformularer, som kræver yderligere sider til at bekræfte status eller resultat for registrering. Fx når en bruger logger på og når man vil se sin profilside. Hvis login oplysingerne er forkerte, kommer man til en side, som oplyser brugeren om fejlen og opfordrer til at prøve igen.

Website navigation diagrammet i giver overblik over alle sider, undersider og links på vores hjemmeside i forbindelse med databasen.



5.2 Sketches, wireframes, og mockups

Sketches

Inden vi begyndte at skabe vores wireframes lavede vi sketches til de vigtigste elementer på vores website. Se bilag 4

Wireframes

Inden vi kunne gå i gang med at lave vores website blev vi nødt til at strukturere dets elementer ved hjælp af wireframes til de forskellige sider. Se bilag 5

Mockups

Vi har lavet mockups af vores website for at hjælpe os selv med at komme frem til det rigtige design, og for at vise hvad vi som udgangspunkt skal gå efter. Se bilag 6

6. Design process

6.1 Logo design

Farver:

ff8a00

555555

Skrifttype:

Cabin

Typeface download link: <http://www.fontsquirrel.com/fonts/cabin>

6.2 Design manual

Farver:

ff8a00

555555

c1c1c1

f7f5f5

Skrifttyper:

Primær:

Helvetica Neue

Sekundære:

Helvetica, Arial, sans-serif

7. Crowdfunding projekt

For at demonstrere hvordan vores crowdfunding system fungerer, har vi også udarbejdet et projekt, både konceptuelt og materialer til databasen, såsom billeder, video, beskrivelse, mm.

Alle billeder er blevet taget af 3 forskellige fotografer med forskelligt kameraudstyr, forskellige lokationer og lyskilder. Mens de fleste billeder er blevet taget med flash og reflektorskærm, har mange også helt naturlige lyskilder.

Redigeringen af lys, farver og kontraster i billederne er foretaget i Lightroom.

Derefter er special effects, såsom at fjerne en finger så alle modeller kun har 4 fingre i stedet for 5, måne og ildfluer er alt sammen billedmanipuleret gennem photoshop.

Videoen er filmet med Canon 60D i forskellige lyssætninger og lokationer, hvilket gør at alle klip ser meget forskellige ud i råformat.

Videoen er derfor redigeret i Adobe Premiere Pro, hvor der, ud over klipning og timing så videoen passer til musikken, er blevet lagt filter over alle klip samt ændret på farver og kontraster så alt passer sammen.



8. Userstory

En user story bliver brugt erhvervsmæssigt til udvikling og forbedring af virksomheden.

Når man skriver en user story, er det vigtigt at man beskriver hvem brugeren er, hvad brugeren gerne vil opnå og hvad der motiverer brugeren.

I dette tilfælde er "brugeren" vores bidragsydere og projekt startere. Derfor er det i vores interesse at brugeren får dækket så mange af sine behov som muligt, så de har lyst til at benytte DoGood til crowdfunding fremover.

I modsætning til en use case, som er en udvidet form af user stories, skal informationerne være kortfattede og simple.

En user story skal ikke skrives for lang og heller ikke for kort, da den kan blive brugt til forbedring af bl.a. virksomhedens database og hjemmesidens indhold.

Vores user stories er skrevet efter en model, som har hjulpet os med at sikre, at vi får de vigtigste informationer med, bl.a. tekniske krav, begrænsninger og forudsætninger.

Modellen vi har benyttet er formuleret således: "Som en **bruger**, vil jeg gerne **mål**, for at **motivation**".

1. Som bruger vil jeg gerne kunne klikke ind under enkelte projekter og se en detaljeret, uddybende beskrivelse.
2. Som bruger vil jeg gerne kunne støtte crowdfunds økonomisk ved hjælp af online betaling.
3. Som bruger vil jeg gerne kunne logge ind og se mine data, hvilke sager jeg har støttet samt fremskridt i støttede sager.
4. Som bruger vil jeg gerne kunne oprette nye crowdfunds på hjemmesiden, som kan ses og støttes af andre brugere.
5. Som bruger vil jeg gerne kunne følge med i egen crowdfund samt andres crowdfunds via info om fremskridt i form af datoer, beløb, mål, mm.

Ud fra disse user stories er det muligt for os at sikre, at brugeren har de nødvendige aktioner tilgængelige fra vores hjemmeside.

Detaljeret beskrivelse af 2 user stories

User story Beskrivelse:

Navn: Scenarie 1

Identifiser: US01

Forudsætninger: Forbindelse til databasen uden login

Grundkursus: US01 starter når brugeren går ind på forsiden
Brugeren klikker på et enkelt projekt.

Alternativ Kursus A: Brugeren søger på projekt og klikker ind på projekt
via søgeresultat.

User story Beskrivelse:

Navn: Scenarie 4

Identifiser: US02

Forudsætninger: Forbindelse til databasen med login
Brugeren skal være logget ind

Grundkursus: US02 starter når brugeren logger ind
Brugeren klikker på "Opret projekt"
Brugeren udfylder informationsformel vedrørende projektet.
Brugeren klikker opret og opretter dermed nyt projekt.

Alternativ Kursus A: Brugeren går ind på kontaktformularen
Brugeren kontakter admin og beskriver projekt
Admin opretter projekt for brugeren.

9. Database opbygning

For at opbygge en database, som kan opfylde de krav, som vi tidligere har bestemt til vores database, startede vi med at finde ud af hvilke oplysninger der skal bruges på hjemmesiden. Herefter fandt vi ud af hvordan vi kunne dele disse oplysninger op i forskellige kategorier, med beskrivelser af de forskellige egenskaber.

Herefter har vi lavet et ER-Diagram, som er en organisering af kategorierne, og som hjælper os til at "reducere og helt fjerne dataredundans, fordi det er utroligt svært at gemme objekter i en relationel database, der har de samme oplysninger flere steder."*5

Denne proces er normalisering af data, og der er flere optimerings niveauer (normal forms) kaldet NF. Der findes 1. NF, 2. NF og 3. NF. Jo større nummeret er, des bedre optimeret er dataskemaet.

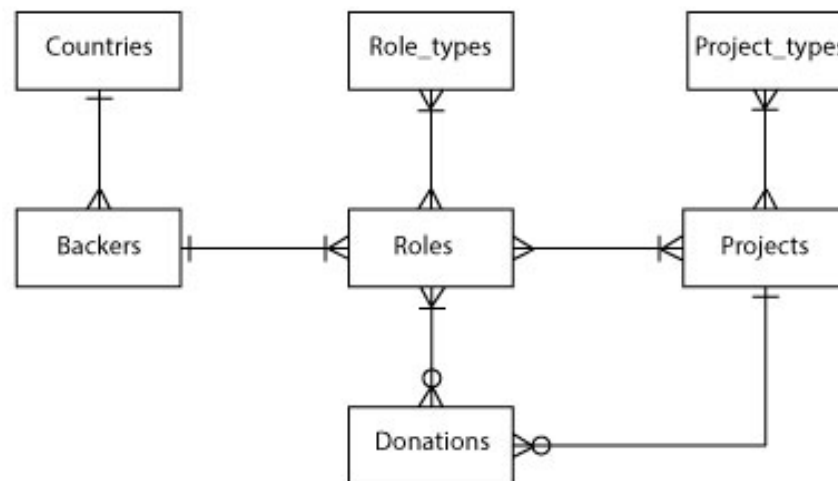
De forskellige kategorier med oplysninger bliver grupperet i tabeller, som vi bruger til at lave ER-Diagrammet med. Vi normaliserer diagrammet således, at det svarer til 3. NF, da dette er den bedste praksis og den løsning vi skal arbejde med i dette projekt.

9.1 ER-diagrammer

Når kravene til vores hjemmeside er fundet, og vi ved hvordan designet skal være, hvilken målgruppe vi arbejder med samt hvad målene er, kan vi gå i gang med at planlægge databasen. Planlægningen består af hvordan databasen skal udformes, hvilke tabeller der skal være i databasen, hvilke relationer der skal være mellem de forskellige tabeller og hvilken slags data der skal være i tabellerne. Måden vi planlægger vores database er ved at bruge ER-Diagrammer.

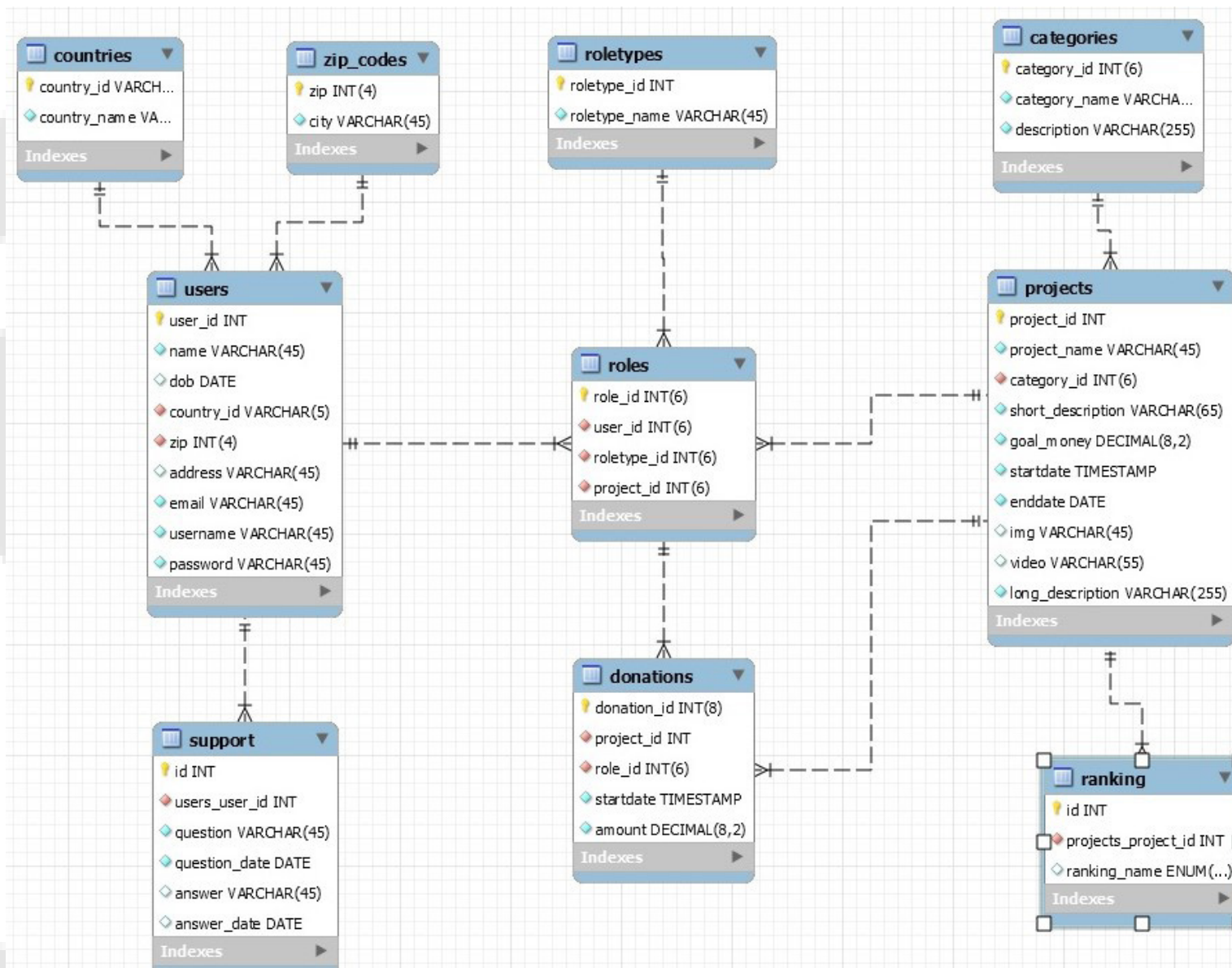
Det første ER-Diagram vi laver, bliver lavet uden brug af programmer. Det laves ved at vi i gruppen diskuterer hvilke ting man skal kunne på hjemmesiden, samt hvilke data man skal bruge. Derefter laves et konceptuelt ER-Diagram, som viser de tanker vi har omkring projektet.

Figur ERConcept - Konceptuel ER-Diagram



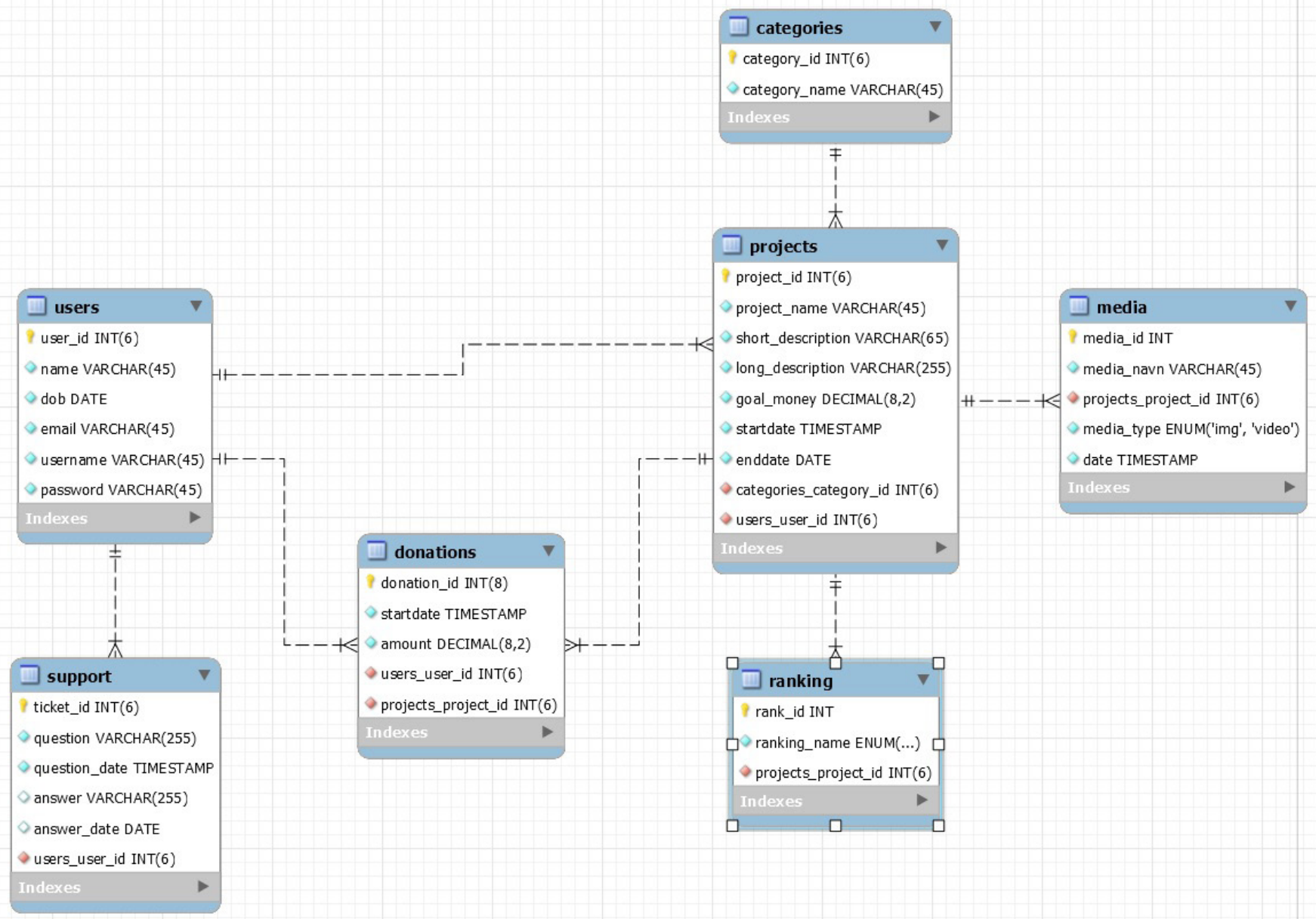
Når vi har lavet det konceptuelle ER-Diagram, arbejdes der i MySQL Workbench, som er det program vi har brugt til at lave databasen til dette projekt. Her bygges de forskellige tabeller, de data der skal bruges i databasen tilføjes, og der defineres hvilke datatyper der er mest relevant til de forskellige entiteter.

Figur ER1 - 1. ER-Diagram



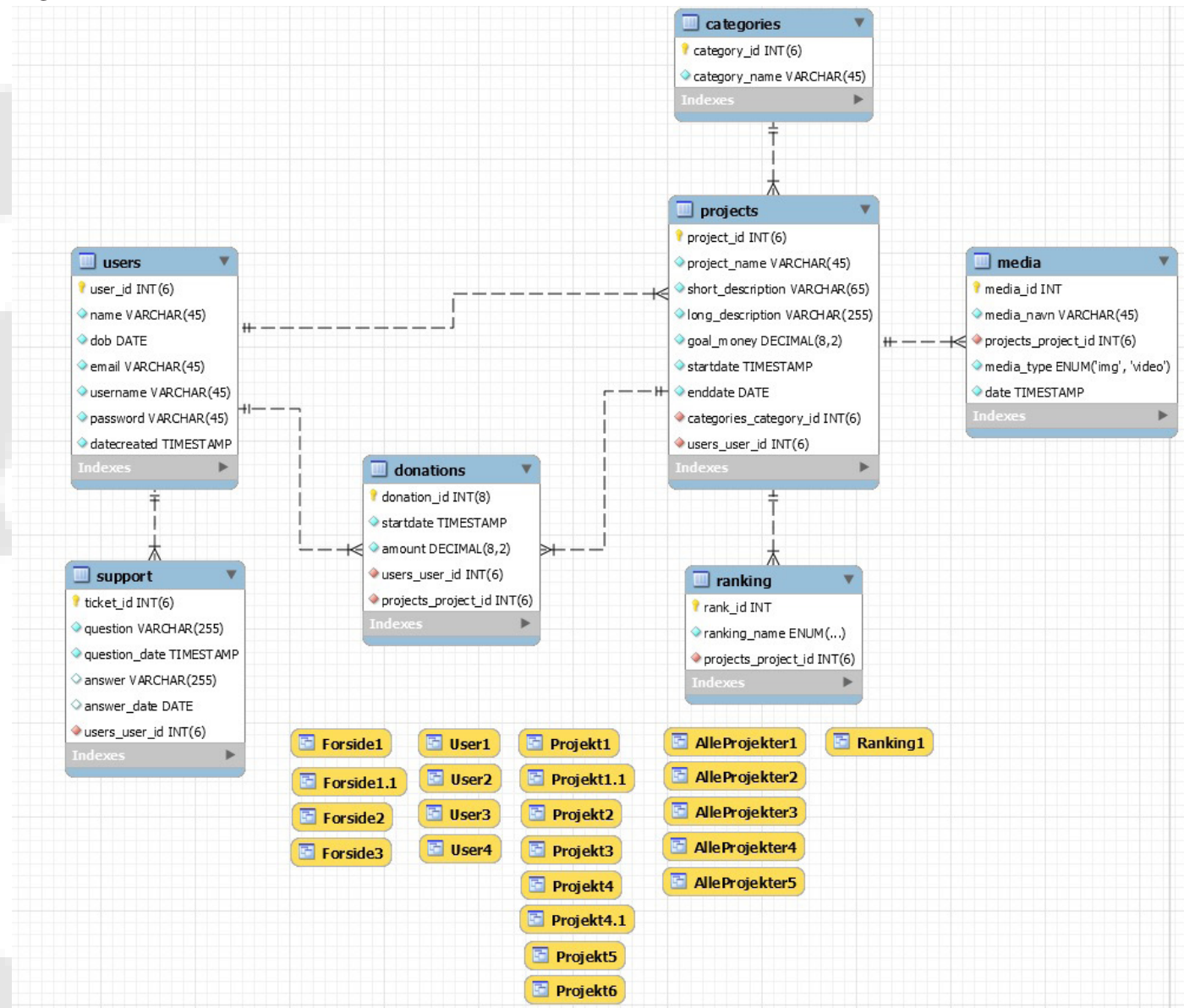
Efterhånden i arbejdsprocessen vil der være nogle krav der ændrer sig. Derfor er der tabeller eller data der tilføjes eller fjernes. I dette tilfælde er vi nået frem til, at vi ikke har behov for at kunne lokalisere brugerne. Derfor har vi fjernet tabellerne Countries og Zipcodes.

Figur ER2 - 2. ER-Diagram



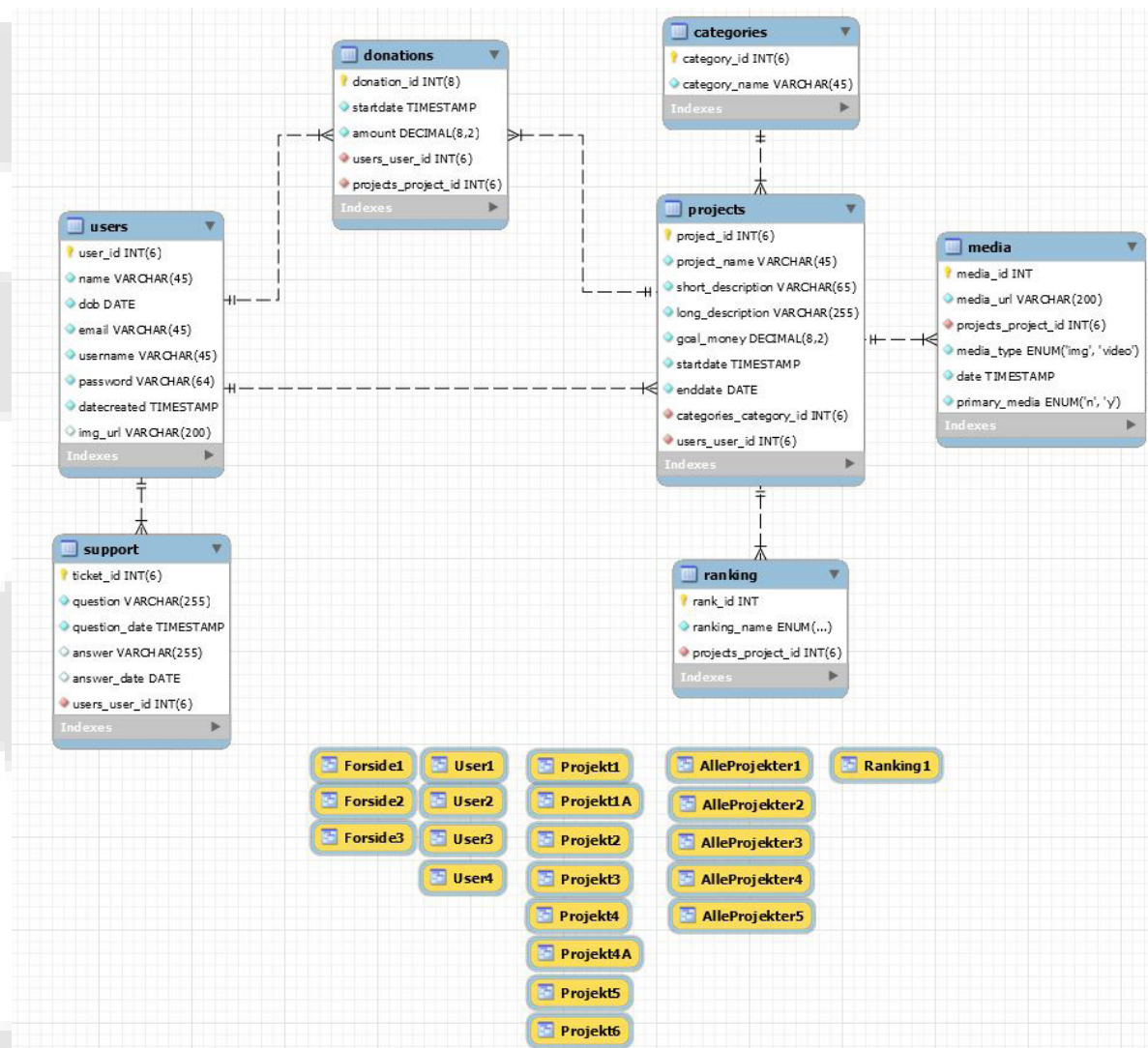
Efterhånden som tabellerne tilføjes/fjernes, indsættes data i tabellerne, for at se tjekke om det fungerer, samt for at kunne se strukturen arbejdet. Afslutningsvis besluttes der, hvilken form for data der skal trækkes ud af databasen og vises på hjemmesiden. Til dette bruges Views, som gennemgås senere.

Figur ER3 – Endelig ER-Diagram med Views



Der kan forekomme ændringer i sidste øjeblik, hvor der laves mindre reguleringer i tabeller, dog uden at den samlede struktur ændres. Det kan ses i dette eksempel, hvor foretages små ændringer fra ER3 og til ER4, som er den endelige version. En af de ændringer der foretages er i tabellen media, hvor der indsættes et ekstra felt med forespørgsel om, hvorvidt billedet/videoen er projektets primære medie, som skal illustrere projektet på hjemmesiden.

Figur ER4 – Last-minute change ER-Diagram med Views



9.2 ER-diagram på 3.NF

Som det kan ses i ER-Final, har vi opdelt vores database i 7 forskellige kategorier. Brugerrelateret, projektrelateret, donationsrelateret, kategorier, medierelateret, rankingrelateret og admin-/supportrelateret.

Til dette har vi oprettet tilsvarende tabeller: users, projects, donations, categories, media, ranking og support, og har her indsat de relevante attributter. Vi har valgt at holde tabelnavne, og attributbetegnelser på engelsk, da dette er et produkt der vil kunne sælges til internationale virksomheder. Derfor vil det være nemmest at lave det hele på engelsk fra start af.

Selve data i databasen er lavet på dansk, da vores prototype af en hjemmeside også er på dansk, og målrettet mod det danske marked.

Som nævnt tidligere, startede vi med at lave et konceptuelt ER-Diagram, og udviklede vores ER-Diagrammer herfra. Her havde vi oversigt over de forskellige relationer mellem de tabeller som vi har lavet.

I det konceptuelle ER-Diagram, var der flere mange-til-mange relationer, og dette svarer ikke til 3.NF. Derfor udviklede vi det, indtil vi fik det endelige ER-Diagram, hvor vi har fjernet tabellerne `role_types` og `roles`, da der imellem disse, samt mellem `roles` og `projects`, var mange-til-mange relationer. Vi har i stedet lavet en direkte relation mellem `users` og `projects`, og efterfølgende lagt en relation fra `users` til `donations`, samt fra `donations` til `projects` for at undgå disse mange-til-mange relationer. Dette gøres for at undgå forvirring i databasen omkring en `users` rolle til forskellige projekter og donationer. Desuden er `project_types` blevet ændret til kategorier, hvor et projekt kun kan indgå under 1 kategori, men 1 kategori kan have mange projekter. Normalvis vil man tilføje flere tabeller, for at optimere på databasen, men i vores tilfælde gav det bedre mening at begrænse tabellerne, og udtænke en anden måde at relatere de forskellige tabeller.

På denne måde har vi normaliseret ER-Final diagrammet til 3.NF. Nedenfor vil vi opstille de forskellige relationer i databasen mellem tabellerne:

- Users til donations -> 1-til-mange relation: 1 user kan have mange donationer, men 1 donation kan kun have 1 user.
- Projects til donations -> 1-til-mange relation: 1 projekt kan have mange donationer, men hver donation kan kun laves til 1 projekt.
- Users til projects -> 1-til-mange relation: 1 user kan have mange projekter, men et projekt kan kun have 1 user (creator).
- Users til support -> 1-til-mange relation: 1 user kan have mange tickets, men en ticket kan kun have 1 user der har oprettet den.
- Categories til projects -> 1-til-mange relation: 1 kategori kan have mange projekter, men 1 projekt kan kun tilhøre 1 kategori.
- Projects til media -> 1-til-mange relation: 1 projekt kan have flere billeder og videoer, men 1 billede eller 1 video kan kun tilhøre 1 projekt.
- Projects til ranking -> 1-til-mange relation: 1 projekt kan få rankings fra mange forskellige brugere, men hver bruger kan kun give 1 ranking til hvert projekt.

Den sidste relation i mellem `projects` og `ranking` er ikke færdigudviklet i vores database endnu. Det vil være således, at hver bruger bliver identificeret på deres IP-Adresse, og hver IP-Adresse vil kun kunne bedømme hvert projekt 1 gang.

9.3 Data til database

Når vi har klargjort strukturen af vores database i ER-Diagrammet ER-Final, kan vi begynde at sætte data ind i de forskellige tabeller. Dette gøres enten via SQL-Queries, eller i vores tilfælde via programmet MySQL Workbench, hvor vi kan sætte data ind inden databasen bliver aktiv. Dette er det samme program, som vi har brugt til at lave vores ER-Diagrammer, og har testet databasen i.

Vi har lavet forskellige testprojekter, og 1 enkelt konkret eksempelprojekt, som hedder ElfQuest. Vi har sørget for at oprette forskellige testbrugere (som i dette tilfælde er alle 5 medlemmer i gruppen), forskellige donationer til de forskellige testprojekter, og generelt sørget for at der er data i alle tabellerne.

I tabellen Categories har vi indsat de kategorier, som vi har planlagt at bruge på vores hjemmeside. I tabellen media, har vi indtastet forskellige URL til billeder og video, som er relateret til de forskellige projekter.

Tabellerne support og ranking er på nuværende tidspunkt kun lavet til fremtidig brug på hjemmesiden, da vi gerne vil udvide hjemmesiden, således at brugere kan give en bedømmelse af de forskellige projekter, og desuden lave support tickets, så de kan få hjælp fra sidens administratorer.

Vi har som tidligere nævnt, sørget for at der er data i alle tabeller, således at vi kan teste både databasen og vores hjemmeside.

9.4 Attributtabel

I attributtabellen vises oplysningerne der findes i ER-Diagram, dog uden beskrivelse af relationer. Attributtabellen benyttes til at planlægge struktur og datatyper til opbygningen af tabellerne i databasen. Dette gøres primært for at skabe overblik og dermed simplificere arbejdet i databasen.

En attributtabel kan ydermere bruges i virksomheden, hvor man let kan skabe sig et overblik over en databasens indhold uden at skulle tilgå databasen. Dette skal især benyttes ved dokumentering, vedligeholdelse og planlægning af fremtidige opdateringer i databasen. Vi har lavet en attributtabel, som viser de tabeller der benyttes i databasen, samt indholdet af disse. Vi har vurderet hvilket indhold der er behov for på hjemmesiden samt hvilke datatyper der er mest relevante i de forskellige tabeller i forhold til brugernes krav.

Som man kan se i bilag 7, findes følgende kolonner:

- Entity (beskriver tabeller i databasen)
- Attributes (beskriver rækkerne i de enkelte tabeller)
- Value (beskriver hvilken værdi der skal indtastes i rækkerne)
- Notes (beskriver specielle ting ved denne række, fx Unique som ikke accepterer at der findes duplikater i rækken)
- Datatype (enten Numerisk eller Alfanumerisk)

9.5 Sql views til queries på hjemmesiden

Der er lavet 22 forskellige Views, som tester forskellige områder, der skal bruges på hjemmesiden, samt hvor på hjemmesiden koden senere skal bruges for at trække data ud af databasen, eller for at gemme data i databasen.

9.5.1 Forside

Den view der er navngivet Forside1 er dataudtræk, som bruges på forsiden af hjemmesiden.

Forside1

I viewet Forside1 udvælges det nyeste projekt i databasen, som skal vises på forsiden af hjemmesiden. Dette gøres ved at sortere projekterne efter oprettelsesdato (startdate) med det nyeste først. Herefter laves en grænse på 1 (limit), som gør at kun det nyeste projekt vises.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`User1` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`User1`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `User1` AS

SELECT name, username
FROM users
WHERE user_id=1;
```

9.5.2 Projekt

Projekt1.A

I det 2. view trækkes alle de medier ud, som er brugt til dette projekt. Dette omhandler de forskellige billeder og videoer, som projektopretteren har lagt links til.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Projekt1A` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`Projekt1A`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `Projekt1A` AS

SELECT media_url
FROM media
WHERE projects_project_id=2;
```


Projekt2

Dette view er lidt mere avanceret end de resterende. I dette view udregnes, hvor mange procent af projektets målbeløb der allerede er doneret. Hvis målet er på 10000, og der er doneret 5000 i alt, regnes dette view ud så der vises 50%. Det viser samtidig hvor meget der er blevet doneret allerede, samt hvor meget målet for projektet er. Dette gøres ved først at lægge alle donationer sammen til det pågældende projekt-id, hvorefter det ganges med 100 og divideres med målet for projektet (goal_money).

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Projekt2` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`Projekt2`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `Projekt2` AS

SELECT SUM(donations.amount) AS total, donations.projects_project_id as probject_id, projects.goal_money, ((SUM(amount)*100)/projects.goal_money) as procent
FROM donations
inner join projects on projects.project_id = donations.projects_project_id
group by donations.projects_project_id;
```

Projekt3

I dette view bruges lidt fra viewet Projekt2, da vi tager summen af alle de donationer, der er i det specifikke projekt og viser denne sum.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Projekt3` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`Projekt3`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `Projekt3` AS

SELECT sum(amount) AS total
FROM donations
WHERE projects_project_id=2;
```

Projekt4

I viewet Projekt4 vises hvor mange unikke brugere, der har doneret til det pågældende projekt. Først trækkes alle brugere der har doneret ud, hvor projekt-id svarer til det valgte projek. Dette bruger vi igen med en SELECT DISTINCT, som sørger for at hver bruger kun bliver vist 1 gang. Til dette bruges COUNT således at der optælles hvor mange der er i alt.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Projekt4` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`Projekt4`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `Projekt4` AS

SELECT COUNT(DISTINCT users_user_id) AS Donation
FROM donations
WHERE projects_project_id=1;
```

Projekt4.A

I dette view vises hvor mange donationer der er foretaget i alt. Her tætages ikke højde for, at en bruger kan have doneret flere gange, så alle donationer tælles med.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Projekt4A` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`Projekt4A`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `Projekt4A` AS

SELECT COUNT(users_user_id) AS Donation
FROM donations
WHERE projects_project_id=1;
```

Projekt5

Dette view er ikke nået at blive lavet, men det skulle have vist, hvor mange dage der var tilbage til projektets deadline. Dette kan laves til fremtidig brug på hjemmesiden.

Projekt6

Dette view viser hvad målet for projektet er. Vi trækker goal_money ud fra tabellen projects.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Projekt6` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`Projekt6`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `Projekt6` AS

SELECT goal_money
FROM projects
WHERE project_id=1;
```

9.5.3 Kategorier

AlleProjekter1, AlleProjekter2, AlleProjekter3, AlleProjekter4 og AlleProjekter5, er views der viser de dataudtræk, der bruges til at trække informationer ud omkring de forskellige projekter, der ligger i hver enkelt kategori på hjemmesiden. Dette bruges til kategorimenuen.

AlleProjekter1

Dette view trækker alle projekter ud, der har categoryid 1, som er Art.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`AlleProjekter1` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`AlleProjekter1`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `AlleProjekter1` AS

SELECT project_name, short_description, long_description
FROM projects, categories
WHERE categories_category_id=category_id
AND category_id=1
ORDER BY project_id;
```

AlleProjekter2

Dette view trækker alle projekter ud, der har categoryid 2, som er Charity.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`AlleProjekter2` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`AlleProjekter2`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `AlleProjekter2` AS

SELECT project_name, short_description, long_description
FROM projects, categories
WHERE categories_category_id=category_id
AND category_id=2
ORDER BY project_id;
```

AlleProjekter3

Dette view trækker alle projekter ud, der har categoryid 3, som er Fashion.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`AlleProjekter3` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`AlleProjekter3`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `AlleProjekter3` AS

SELECT project_name, short_description, long_description
FROM projects, categories
WHERE categories_category_id=category_id
AND category_id=3
ORDER BY project_id;
```

AlleProjekter4

Dette view trækker alle projekter ud, der har categoryid 4, som er Food.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`AlleProjekter4` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`AlleProjekter4`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `AlleProjekter4` AS

SELECT project_name, short_description, long_description
FROM projects, categories
WHERE categories_category_id=category_id
AND category_id=4
ORDER BY project_id;
```

AlleProjekter5

Dette view trækker alle projekter ud, der har categoryid 5, som er Sport.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`AlleProjekter5` ;
DROP TABLE IF EXISTS `annastacy_dogood`.`AlleProjekter5`;
USE `annastacy_dogood`;
CREATE OR REPLACE VIEW `AlleProjekter5` AS

SELECT project_name, short_description, long_description
FROM projects, categories
WHERE categories_category_id=category_id
AND category_id=5
ORDER BY project_id;
```

9.5.4 Ranking

Ranking1 er det view, som viser dataudtræk der bliver brugt til at lave en gennemsnitsrangering af hvert enkelt projekt.

Ranking1

Dette tager alle tal i ranking_name, som svarer til det specifikke projekt-id, og viser gennemsnitsværdien ved hjælp fra SELECT AVG.

```
DROP VIEW IF EXISTS `annastacy_dogood`.`Ranking1` ;  
DROP TABLE IF EXISTS `annastacy_dogood`.`Ranking1` ;  
USE `annastacy_dogood` ;  
CREATE OR REPLACE VIEW `Ranking1` AS  
  
SELECT AVG(ranking_name) AS average_ranking  
FROM ranking  
WHERE projects_project_id=1;
```

Denne view har vi ikke brugt til hjemmesiden dog pga. mangel af tid.

10. Interactions udvikling (kodning)

Når vi har forberedt al informationsarkitektursmodeller, website navigation diagram og databasen med sql-forespørgsler, kan hjemmesiden kodes. Vi har benyttet DreamWeaver til dette formål.

10.1 Responsive design og Templates

Vi har anvendte responsive grid kode struktur, som hedder simplegrid fra <http://thisisdallas.github.com> . Det hjælper med at sætte struktur på sider div- elementer ved hjælp af 12-grid kolumner, som bruger tidligere defineret classes, fx:

```
col-2-12 + col-8-12 + col-2-12 = col-12-12
```

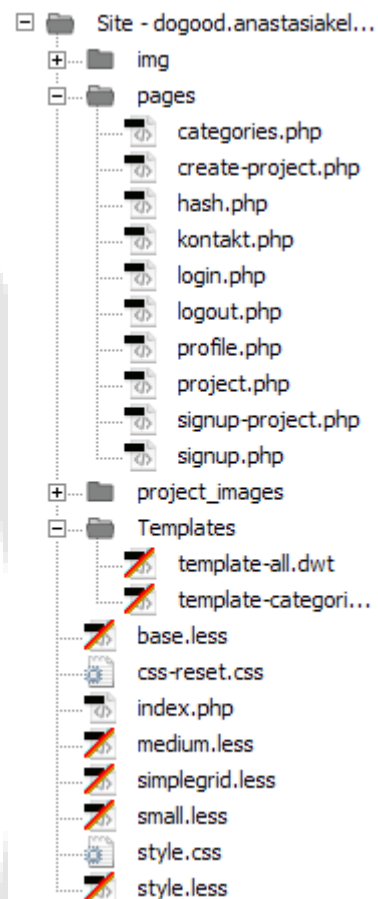
Det gør det nemmere at style siderne bagefter, især når man skal også style siderne i ipad og mobil størrelser. Det gør man ved at tildele classes til eksisterende kolumner, fx. hide-on-mobile, vis-on-medium osv. I dette projektet havde vi ikke tid nok at anvende responsive styling, men det kan let udføres senere, hvis nødvendigt.

Så anvendte vi Templates for at lave sideres skelet, som kunne genbruges til flere sider senere. Templates gemmer uforanderlige dele af hjemmesiden, som primær menu og footer. Det er rigtig praktisk, så man undgår at kopier koden til andre sider igen og igen, samt undgår eventuel sletning af nogle vigtige dele a koden.

Vi har lavet 2 template sider, fordi der er 2 typer sider på vores hjemmeside: med en side navigation bar til venstre, og uden den, som også kan ses i Wireframes.

10.2 Kode struktur

Efterfølgende opbyggede vi hjemmesidens struktur som vist i Website navigation diagram. Kode struktur er vist nedenunder:



Less filer er skabt ved hjælp af program Winless, som er en del simplegrid løsning, hvor alle less filer til forskellige sider er opsamlet i en eneste style.css fil. Det gør man når der er for mange sider og det bliver svært at håndtere så mange enklere .css filer. Desuden anvender simplegrid koden også .less filer.

10.3 PHP

PHP-koden bruges til forskellige formål på siderne:

- Tilslutning til MySQL-database, der indeholder tabeller med data:

```
<?php session_start();
// connection using mysqli - changing between remote and local server

// Remote server
//define("HOSTNAME","mysql9.gigahost.dk");
define("MYSQLUSER", "annastacy");
define("MYSQLPASS", "");
define("MYSQLDB", "annastacy_dogood");

// Localhost server
define("HOSTNAME","localhost");
//define("MYSQLUSER", "annastacy");
//define("MYSQLPASS", "");
//define("MYSQLDB", "annastacy_dogood");

$connection = new mysqli(HOSTNAME, MYSQLUSER, MYSQLPASS, MYSQLDB);

if($connection->connect_error){
    die($connection->connect_error);
}
```

- Hentning af billede eller video for specifikt projekt:

```
//dynamic rendering of the database array using a while loop
while($stmt->fetch())
{
?>
    <div class='media-container'>
        <a href="project.php?id=?php echo $project_id ?>">
<?php
    if ($media_type == 'img') {
?>
        '" />
<?php
    } else if ($media_type == 'video') {
        // finds the video ID, e.g. '8eGqmyZydUs' in 'https://www.youtube.com/watch?v=8eGqmyZydUs'
        preg_match('/[\\?\\&]v=([^\?\\&]+)/', $media_url, $matches);
        $id = $matches[1];
        // find out the image thumbnail for the video...
        $media_url = "http://img.youtube.com/vi/" . $id . "/mqdefault.jpg";
?>
        '" />
<?php
    }
}
?>
```

- Til at lave oversigt over projekter i databasen, baseret på kategorier og forskellige efterspørgsler:

```
if(isset($category_id)) {  
  
    $stmt = $connection->prepare("SELECT project_id, project_name, name, media_url, media_type, startdate,  
short_description FROM projects, media, users WHERE projects_project_id = project_id AND user_id = users_user_id  
AND primary_media = 'y' AND categories_category_id = ? ORDER BY startdate DESC LIMIT $start_from,  
$num_rec_per_page;");  
    $stmt->bind_param("i", $category_id);  
} else {  
    $stmt = $connection->prepare("SELECT project_id, project_name, name, media_url, media_type, startdate,  
short_description FROM projects, media, users WHERE projects_project_id = project_id AND user_id = users_user_id  
AND primary_media = 'y' ORDER BY startdate ASC LIMIT $start_from, $num_rec_per_page;");  
}  
$stmt->bind_result($project_id, $project_name, $name, $media_url, $media_type, $startdate, $short_description);  
$stmt->execute();
```

- Til at skabe en menu i venstre side med projektkategorier og fremhævelse af menupunktet for den aktive side:

```
<?php  
if(isset($_GET["id"]))  
    $category_id = $_GET["id"];  
  
// new query  
$categories = $connection->query("SELECT category_id, category_name FROM categories ORDER BY category_name;");  
  
// dynamic rendering of the database array using a while loop  
while($row = $categories->fetch_array())  
{  
?  
    <li><a class="category <?php if($row[ 'category_id' ] == $category_id) echo "selected"; ?>" href=  
"../pages/categories.php?id=<?php echo $row[ 'category_id' ]; ?>">  
    <?php echo $row[ 'category_name' ]?></a>  
?  
}<?php  
}
```

- Hentning af statistiske data:

```
<?php
// new query for getting project_id
if(isset($_GET["id"]))
    $project_id = $_GET["id"];

$stmt = $connection->prepare("SELECT project_name, name, media_id, media_url, media_type, projects.startdate,
projects.enddate, short_description, long_description, goal_money, sum(amount) AS donations_sum, count(distinct
donations.users_user_id) AS donations_count FROM projects, media, users, donations
WHERE media.projects_project_id = project_id AND donations.projects_project_id = project_id AND user_id =
donations.users_user_id AND primary_media = 'y' AND project_id = ?");
if (!$stmt) {
    die($connection->error);
}
$stmt->bind_param("i", $project_id);
$stmt->bind_result($project_name, $name, $media_id, $media_url, $media_type, $startdate, $enddate,
$short_description, $long_description, $goal_money, $donations_sum, $donations_count);
$stmt->execute();
if($stmt->fetch()) {
    $percent_funded = round($donations_sum / $goal_money * 100, 1);
    $today = new DateTime("now");
    $goal_money = round($goal_money);
    $donations_sum = round($donations_sum);
    $days_left = $today->diff(new DateTime($enddate))->days;
}
?>
```

- Visning af en integreret YouTube-afspiller, når brugeren klikker projektets thumbnail under siden "Alle projekter":

```
<?php
  if ($media_type == 'img') {
?>
    <a href="media.php?id=<?php echo $media_id ?>"
      ' " />
    </a>
<?php
  } else if ($media_type == 'video') {
    // finds the video ID, e.g. '8eGqmyZydUs' in 'https://www.youtube.com/watch?v=8eGqmyZydUs'
    preg_match('/[\\?\\&]v=([^\?\\&]+)/', $media_url, $matches);
    $id = $matches[1];
    // find out the image thumbnail for the video...
    $media_url = "https://www.youtube.com/embed/" . $id .
"?rel=0&showinfo=0&color=white&iv_load_policy=3";
?>
<!-- html and style classes are borrowed from http://embedresponsively.com/ -->
    <div class='full-size-video'>
      <div class='embed-container'><iframe src="<?php echo $media_url ?>" style="border: none;"
allowfullscreen></iframe></div>
      </div>
    </div>
  </div>
<?php
  }
?>
```

Sideinddeling og sidenummerering (pagination), mens man søger efter oplysninger om projekter:

```
<?php
// BEGIN pagination code from
http://runnable.com/U8dzQWEzMxxqeQ_E/php-pagination-example-using-mysql-database-for-dbms
$num_rec_per_page=6;
if (isset($_GET["page"])) { $page = $_GET["page"]; } else { $page=1; };
$start_from = ($page-1) * $num_rec_per_page;
// END pagination code from
```

```
<?php
// BEGIN pagination code from
http://runnable.com/U8dzQWEzMxxqeQ_E/php-pagination-example-using-mysql-database-for-dbms

if(isset($category_id)) {
    $stmt = $connection->prepare("SELECT count(*) FROM projects WHERE categories_category_id = ?;");
    $stmt->bind_param("i", $category_id);
} else {
    $stmt = $connection->prepare("SELECT count(*) FROM projects;");
}
$stmt->bind_result($project_count);
$stmt->execute();
$stmt->fetch();
$total_records = $project_count;
$total_pages = ceil($total_records / $num_rec_per_page);

// _PHP_SELF was undefined, found solution here:
http://stackoverflow.com/questions/12710803/undefined-variable-php-self
$scriptName = $_SERVER['SCRIPT_NAME'];

echo "<a href='$scriptName?page=1'>."|&lt;.'"</a> "; // Goto 1st page

for ($i=1; $i<=$total_pages; $i++) {
    echo "<a href='$scriptName?page=".$i.'">".$i."</a> ";
};

echo "<a href='$scriptName?page=$total_pages'>."&gt;|"</a> "; // Goto last page

// END pagination code from
http://runnable.com/U8dzQWEzMxxqeQ_E/php-pagination-example-using-mysql-database-for-dbms
?>
```

Til at gøre det muligt for brugeren at indsætte oplysninger idatabasen ved registrering vha. Post-metode, samt at vise fejlmeldinger, hvis dette ikke fungerer:

```
<?php
//print_r($_POST);

// prepare data for insertion into database
if (isset($_POST['submit'])) {
    // collect form values
    $name = $_POST['name'];
    $email = $_POST['email'];
    //adding password encryption
    $password = hash("sha256", $_POST['password']);
    // check if username and email exist, else insert data into database
    $exists = 0;
    $check = $connection->query("SELECT email from users WHERE email = '$email' LIMIT 1");
```

```
    if ($check->num_rows > 0) {
        echo "<p>E-mail er allerede registreret!</p>";
    } else {

        #####
        # insert data into mysql database #
        #####

        //sql query inserting values into database
        $sql = "INSERT INTO users (email, password, name) VALUES ('$email', '$password', '$name')";

        //execute the query

        if($connection->query($sql){
            header("Location: login.php");
        } else {
            echo "Beklager, vi kunne ikke oprette din bruger";
        }
    }
}

} //end if isset

?>
```

Samt mange andre ting, som kan ses i den vedhæftede zip-mappe "Kode".

10.4 Login system

Loginsystemet er lavet som en meget basal login/user løsning.

Der findes 3 formularer på vores hjemmesiden som skal fungere i forbindelse med log-in systemet.

- Signup
- Login
- Logout

Både signup og login sider fører til Profil siden. Kun brugeren har adgang til deres profilside, hvilket er løst vha. en restricted-fil. Vi "hash-er" adgangskoderne, så de er bedre sikret.

Signup og login bruger begge users-tabellen i databasen. Signup giver kun mulighed for at sende data til databasen, hvor login giver mulighed for at tjekke for eksisterende data i databasen.

Signup fungerer som registrering og tilgås fra en hvilken som helst side på hjemmesiden. Brugeren indtaster de vigtigste oplysninger, som navn, e-mail og kode, hvorefter de kan registrere deres bruger. Efter registreringen er det muligt at logge ind og interagere med siden som bruger. På nuværende tidspunkt er "start projekt" ikke eksisterende som funktion, og brugeren må i stedet kontakte admin for at oprette et nyt projekt i første omgang.

mockup-sider (betalings formular)

11. Konklusion og refleksion

Når vi ser på vores problemformulering, og vurderer hvorvidt vi har fået lavet det vi skulle, er der både plusser og minusser.

”Formålet med projektet er at skabe en prototype til en crowdfunding hjemmeside. Prototypen skal fungere med database, og give brugere mulighed for at se, støtte og oprette projekter via siden.”

Vi kan spørge os selv, har vi nået vores mål? Vi har lavet en crowdfunding side der fungerer sammen med databasen. Dog er muligheden for at støtte og oprette projekter temmelig begrænset. Vores hjemmeside giver brugerne mulighed for at se de forskellige projekter, for at oprette en profil på siden samt mulighed for at logge ind på siden og se deres brugerprofil. Her kan brugeren se hvilke projekter vedkommende har støttet, og hvilke de har oprettet. Den giver endnu ikke mulighed for at donere penge til de forskellige projekter, da dette kræver en betalingsformel, som skal udarbejdes i samarbejde et firma specialiseret i dette. Ligeledes skal man have fat i sidens administrator for at kunne oprette nye projekter.

Vi har fået lavet en hjemmeside og en database, hvor der er gode fremtidsmuligheder, hvis man har tiden til at udvikle de små mangler som hjemmesiden ender med at have, grundet opgaver der ikke er nået løst.

For at konkludere på hele projektet, så har vi dog lavet en fungerende prototype, der kan vise ideen med hjemmesiden, og mulighederne der er i den.

Refleksioner ifht. projektarbejdet er at vi skal blive bedre til at strukturere vores tid, at uddelegere opgaver mere ligeligt mellem gruppens medlemmer, så medlemmer i gruppen kan arbejde hjemmefra ved fx sygdom. Vi bør strukturere vores arbejde sammen i gruppen, således tiden benyttes så effektivt som muligt, og der er fuldkommen styr på hvem der står for hvad.

12. Referencer

1:

Brainstorm chart er lavet vha. www.bubbl.us

2:

Kilde: <http://www.urbandictionary.com/define.php?term=fundies>

3:

Kilde: <http://www.dr.dk/nyheder/viden/tech/nu-kan-alle-samle-penge-ind>

4:

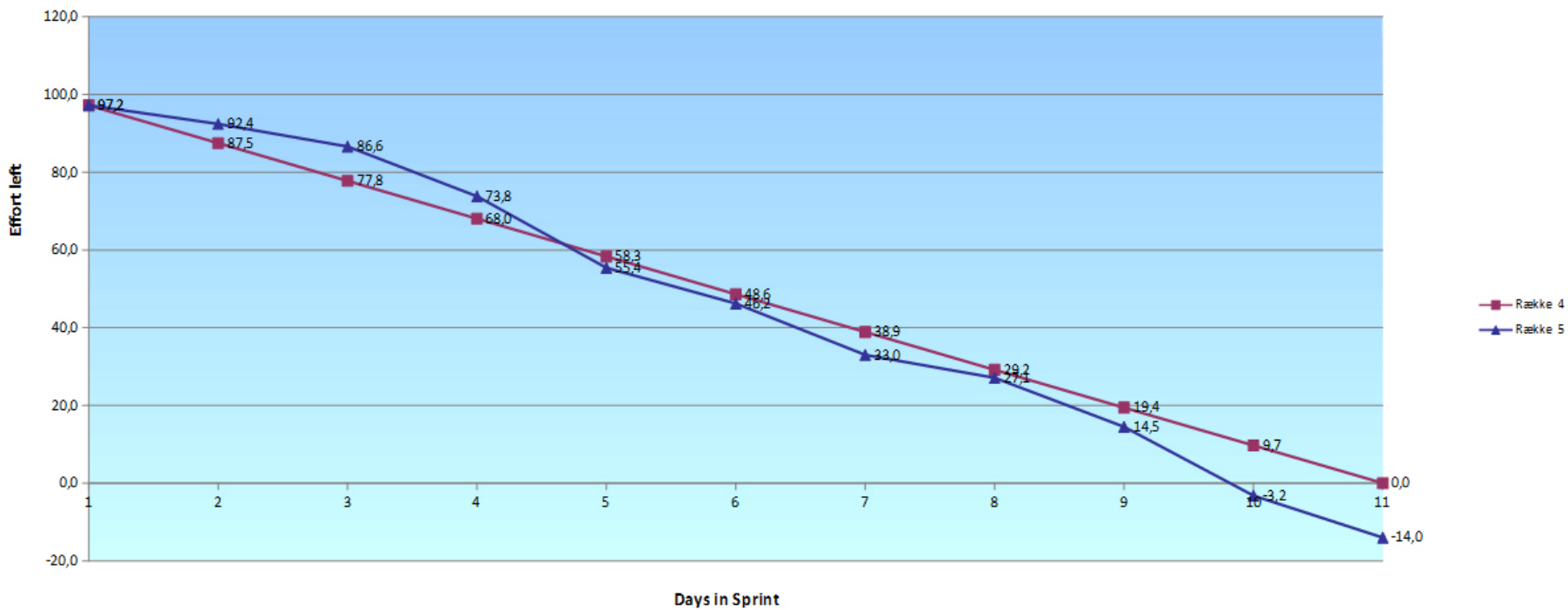
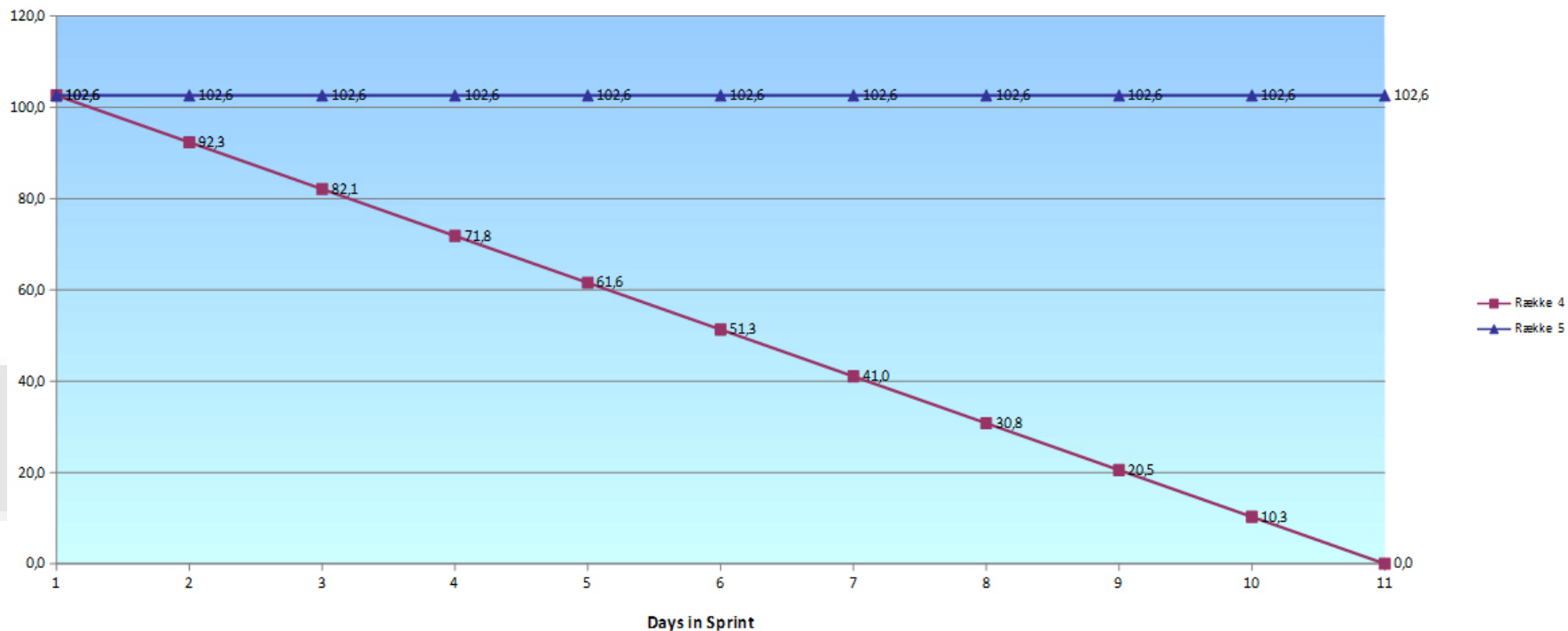
Kilde: "29 Spørgsmål: en moderne retorik til planlægning af kommunikation" af Jan Krag Jacobsen, 2011, s. 17

5:

<http://agiledata.org/essays/dataNormalization.html> "Introduction to Data Normalization: A Database "Best" Practice" af Scott Wybler, 2013

Sprint		Effort "Day 0" calculated using three time estimates (or PERT)							85,4	
Story ID	Stories / Tasks	Planned	No. of team member per task	Total amount of work	Read note i cells for calculations					No. of Team members:
Task ID					A	B	C	M	S _e	Daily working hours:
		Estimate effort in hours							No. of Sprint Days:	
									Available hours of work (resources):	
1	Database									
	1.1 ER-diagram		1	3	3	3	3	3	0	
	1.2 Kommunikation mellem database og hjemmeside		2	12,4	5	6	8	6,2	0,6	
2	Website									
	2.1 Design		2	10,4	3	5	8	5,2	1	Duration >10%- Break down and reestimate
	2.1.1 Mockup og moodboard		2	2,4	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	2.1.2 Logo		1	2,4	2	2	4	2,4	0,4	Duration >10%- Break down and reestimate
	2.2 Kodning		2	18	8	9	10	9	0,4	
				0				0	0	
3	Rapport			0				0	0	
	3.1 Forside		1	1,2	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	3.2 Indholdsfortegnelse		1	1	1	1	1	1	0	
	3.3 Concept		4	12	2	3	4	3	0,4	Duration >10%- Break down and reestimate
	3.4 Rapport Layout		1	3	2	3	4	3	0,4	Duration >10%- Break down and reestimate
	3.6 Konklusion		4	4	1	1	1	1	0	
	3.8 Kodning af database og hjemmeside		2	8	3	4	5	4	0,4	
	3.9 User stories		1	1,2	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	3.10 SCRUM		3	3,6	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	3.11 Design brief		2	2,8	1	1	3	1,4	0,4	Duration >10%- Break down and reestimate

Sprint		Effort "Day 0" calculated using three time estimates (or PERT)							97,2	
Story ID	Stories / Tasks	Planned	No. of team member per task	Total amount of work	Read note i cells for calculations					No. of Team members:
Task ID					A	B	C	M	S _e	Daily working hours:
		Estimate effort in hours							No. of Sprint Days:	
									Available hours of work (resources):	
	ER-diagram		1	2	1	2	3	2	0,4	Duration >10%- Break down and reestimate
	Kommunikation mellem database og hjemmeside		2	12,4	5	6	8	6,2	0,6	
	Views		2	4,8	2	2	4	2,4	0,4	Duration >10%- Break down and reestimate
				0				0	0	
	Website design		2	10,4	3	5	8	5,2	1	Duration >10%- Break down and reestimate
	Mockup og Moodboard		2	2,4	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	Logo		1	2,4	2	2	4	2,4	0,4	Duration >10%- Break down and reestimate
	Kodning		2	18	8	9	10	9	0,4	
				0				0	0	
	Rapport forside		1	1,2	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	Indholdsfortegnelse		1	1	1	1	1	1	0	
	Concept		4	12	2	3	4	3	0,4	Duration >10%- Break down and reestimate
	Rapport layout		1	3	2	3	4	3	0,4	Duration >10%- Break down and reestimate
	Konklusion		4	4	1	1	1	1	0	
	Reflektion		4	4	1	1	1	1	0	
	Beskrivelse af kodning af database og hjemmeside		2	8	3	4	5	4	0,4	
	Userstories		1	1,2	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	SCRUM		3	3,6	1	1	2	1,2	0,2	Duration >10%- Break down and reestimate
	Design brief		2	2,8	1	1	3	1,4	0,4	Duration >10%- Break down and reestimate
	Metode og teknik		4	4	1	1	1	1	0	



Functionality (Feature set, capabilities, generality, security)

Hjemmesiden skal fungere som en skabelon til en hjemmeside med crowdfunding-projekter med minimum et detaljeret eksempel på et crowdfunding-projekt.

Hjemmesiden skal fungere i de større browsere: Chrome, Safari, Firefox, Internet Explorer (Edge).

Hjemmesiden skal give brugeren mulighed for at finde informationer om projekter, projektets formål og videre beskrivelser.

Hjemmesiden skal have mulighed for brugeren kan bidrage økonomisk til sagen via en sikker betalingsløsning (authentication).

Hjemmesiden skal give brugeren mulighed for at følge indsamlingen fremover.

Hjemmesiden skal give folk lyst til at donere vha. tiltalende materiale. (fx billeder, video)

Der skal være mulighed for at printe fra projekt-siden (fx projektbeskrivelse, faktura, osv.)

Usability (User interface issues: human factors, accessibility, aesthetics and consistency)

Hjemmesiden skal have alle vigtige informationer tilgængelige som tekst! (billeder skal indeholde alternativ beskrivelse)

Hjemmesidens navigation skal være simpel og let at sætte sig ind i.

Hjemmesidens tekst/billeder/mm. skal være lette at forstå, og må ikke være misvisende i forhold til websitets formål.

Design på hjemmesiden skal tage højde for informations hierarki principper - både i udseende og opsætning.

Statistiske oplysninger og opdatering omkring dem skal vises på en interessant, tydelig og overskuelig måde.

Hjemmesidens layout og design skal tilrette sig en rigtig målgruppe både æstetisk og indholdsmæssigt.

Printervenlig opsætning.

Reliability (Availability, accuracy, recoverability)

Alle oplysninger og statistikker skal være troværdige (baseret på pålidelige kilder)

Alle oplysninger og statistikker (projektforløb, betalingsstatus) skal opdateres regelmæssigt, sådan at de fremstår troværdigt og korrekt

Forældede oplysninger skal opdateres eller fjernes

Upassende oplysninger skal fjernes eller kontrolleres (af organisationer)

Hvis serveren går ned, skal problemet behandles omgående

Alle vigtige oplysninger skal have back-up

Performance (Throughput, response time, recovery time, start-up time)

Hjemmesiden skal være tilgængelig 24 timer i døgnet, med undtagelse af vedligeholdelsestimer, som skal annonceres i forvejen.

Hjemmesiden opdateres i offline mode

Hvis siden bryder ned, skal den være oppe igen i løbet af 1 time. Systemet skal selv kunne give besked til administratorer, som så kan få siden op igen hurtigst muligt. Når systemet oplever problemer, sendes en alarm til admin. Admin undersøger systemet og tager beslutninger omkring, hvad der skal ske (reboot eller lignende) Siden skal herefter køre igen.

Responstid skal være således at brugeren oplever en god og hurtig hjemmeside, så det ikke bliver en negativ oplevelse at tilgå hjemmesiden. Det ville være relevant med tests af siden funktionalitet for at sikre at brugeren får en god oplevelse.

Start-up tid bør ikke være særlig lang. Hvis en server (webserver / sql server) skal genstartes, bør siden være oppe igen i løbet af 10 minutter, således at der undgås lang tid hvor siden er dysfunktionel. Desuden bør der være noget redundans, således at vi undgår nedetid på siden, hvis en server skal opdateres eller lignende.

Supportability (Testability, adaptability, maintainability, compatibility, configurability, installability, scalability and localizability)

Hjemmesiden skal opdateres løbende med nye oplysninger om projekter, nyheder og nye funktioner når det er relevant.

Hjemmesidens design skal opdateres, hvis hjemmesidens målgruppe eller hjemmesidens mission/vision ændres, således at målgruppen rammes bedst muligt.

Localization - systemet kan have mulighed for at kunne køre på flere sprog (engelsk og dansk)

Systemet skal understøtte mulighed for at flere brugere kan udnytte det samtidig uden at miste performance effektivitet

Systemet skal støtte tilstrækkelig datamængde (scalability)

Design requirement (Constrains the design)

A relational database is required.

Hjemmesidens layout og design skal tilpasses til målgruppen, både æstetisk og indholdsmæssigt.

Design på hjemmesiden skal tage højde for informations hierarki principper - både i udseende og opsætning.

Hjemmesiden skal give folk lyst til at donere vha. tiltalende materiale. (fx billeder, video, stor Støt knappen, osv.)

Oplysninger på hjemmesiden skal vises på en interessant, tydelig og overskuelig måde.

Hjemmesidens navigation skal være simpel og let at følge med.

Alle formular (betaling, registrering, osv.) skal have simpel, brugervenlig design med mindst en visuel identitets tegn på (logo, farve, skrifttype, el.lign.)

Implementation req. (Constrains the coding or construction, E.g. required standards, platform or implementation language)

Hjemmesiden bliver udviklet i PHP og databasen bliver en MySQL database.

Hjemmesiden skal fungere på de større browsere (Chrome, Safari, Firefox, Internet Explorer (Edge)

Hjemmesiden kommer til at fungere og blive testet til Microsoft Windows klienter og MAC OSX klienter.

Hjemmesiden skal være en effektiv hjemmeside, som ikke optager for mange ressourcer på brugerens maskine. Dvs. ikke et for stort ram forbrug, når brugeren browser på siden.

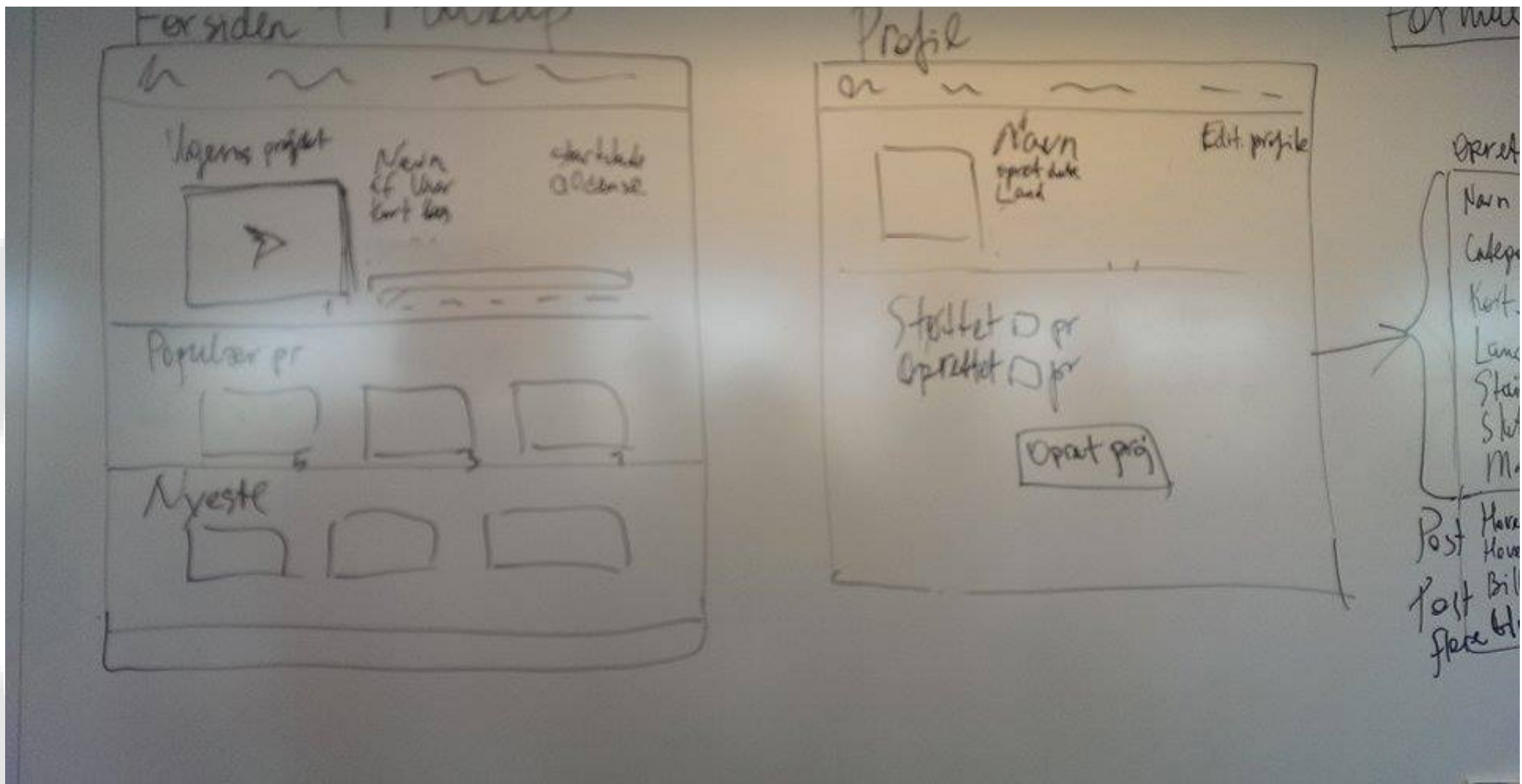
Desuden skal der tages kontakt til online betaling, for at finde ud af hvilke sikkerheds procedurer der er angående dette.

Interface requirement (A requirement to interact with an external item)

Hjemmesiden bliver placeret på en webserver, som ligeledes skal have adgang til en SQL server. Disse skal være på samme lokation / muligvis samme server, for at forhindre en for lang behandlingstid, når der skal behandles data.

Physical requirement

none



Alle projekter

Hand-drawn sketch of a project list interface. At the top, there are three circular icons. Below them is a table with four rows and three columns. The rows are labeled on the left: 'Kalle', 'Kurs', 'Med', 'Med', and 'Sport'. The first cell of the 'Kalle' row contains a box with an 'X' and the text 'Navn af User Kort_b'. The other cells contain question marks. At the bottom of the table is a '<1...>' indicator.

Kalle	[X] Navn af User Kort_b	?	?
Kurs			
Med			
Med			
Sport	?		

<1...>

Projekt-side

Hand-drawn sketch of a project details page. At the top, there are two circular icons. Below them is a large box containing a right-pointing arrow. To the right of this box are labels 'Navn af User' and 'Kort_besk.' with a 'stop' button. Below these is a progress bar with 'start' and 'shut down' labels. Underneath the progress bar are labels '% x kr.', 'x dages dage', and 'mål kr' with values '50' and '5'. At the bottom, there are several horizontal lines and a grid of six square boxes.

Navn af User Kort_besk. stop

start shut down

% x kr. x dages dage mål kr
50 5

Formularer:

zip
by

Opret projekt

Navn <input type="text"/>
Category
Kort_Beskr
Lang_Beskr
Startdate
Slutdate
Mail

Post Hoved_bld
Post Hoved_ud
Post Billeder
Post face_bld. <input type="button" value="Opret"/>

Opret profil

Navn
username
passw
retype passw
e-mail
Zip <input type="text"/>
<input type="button" value="Sign-up"/>
Har du konto?
<input type="button" value="Login"/>

email <input type="text"/>
passw <input type="text"/>
<input type="button" value="Log mig ind"/>


Kontakt os

<input type="text"/>
<input type="button" value="Submit"/>



DOGOOD [HOME](#) [CATEGORIES](#) [ABOUT](#) [GET STARTED](#)

Featured Project

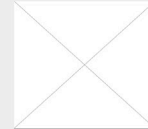
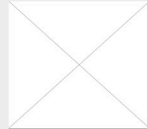
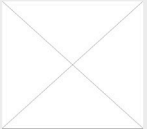


EifQuest - a fan movie
by Babare Inc. [# Film and Video](#)

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corpora suscipit laborio-

64% Funded [❤️](#) **19 Days left**

Popular Projects

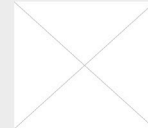
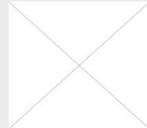
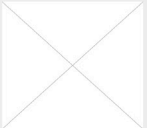


Project Title
Project Author [# Film and Video](#)
33% Funded [❤️](#) **36 Days left**

Project Title
Project Author [# Film and Video](#)
33% Funded [❤️](#) **36 Days left**

Project Title
Project Author [# Film and Video](#)
33% Funded [❤️](#) **36 Days left**

New Projects



Project Title
Project Author [# Film and Video](#)
33% Funded [❤️](#) **36 Days left**

Project Title
Project Author [# Film and Video](#)
33% Funded [❤️](#) **36 Days left**


Project Title
Project Author [# Film and Video](#)
33% Funded [❤️](#) **36 Days left**

[FAQ](#) [Press](#)
[About](#) [Job opportunities](#)
[Careers](#) [API](#)
[Contact](#) [Social links](#)
[Browse Projects](#)

DOGOOD

LOGO NAV1 NAV2 NAV3 SEARCH ... GET STARTED

Featured Project




Smart Watch
Project Author

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?


Button

Popular Projects



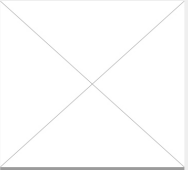
Project Title
Project Author

Button



Project Title
Project Author


Button



Project Title
Project Author


Button

New Projects




Project Title
Project Author

Button



Project Title
Project Author

Button



Project Title
Project Author

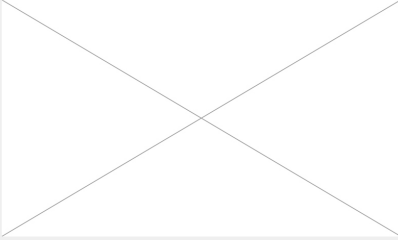
Button

Text Text Social links

LOGO

LOGO NAV1 NAV2 NAV3 SEARCH ... GET STARTED

Project Name



353
backers
60%
pledged
4
days to go


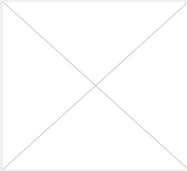

BACK THE PROJECT

Button

Description

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Project Gallery



Text Text Social links

LOGO

Entity	Attributes	Value	Notes	Datatype (N/AN)
Users	user_id	1-X	Max 6 numbers, Unique and Auto Increment	N
	name	a-Å	Max 45 characters	A/N
	dob	date	Birthday of user	N
	email	All Characters	Max 45 characters and Unique	A/N
	username	All Characters	Max 45 characters and Unique	A/N
	password	All Characters	Max 45 characters	A/N
	datecreated	date	DATE, when the profile is created (CURRENT TIMESTAMP as default)	N
	img_url	a-Å	Max 200 characters	A/N
Projects	project_id	1-X	Max 6 numbers and Auto Increment	N
	project_name	a-Å	Max 45 characters	A/N
	short_description	All Characters	Max 65 characters	A/N
	long_description	All Characters	Max 255 characters	A/N
	goal_money	1-X	Max 8 numbers before comma, and Max 2 numbers after comma	N
	startdate	date	DATE, when project is created (CURRENT TIMESTAMP as default)	N
	enddate	date	DATE, which the user chooses, when the project collection ends	N
	categories_category_id	1-X	Max 6 numbers	N
users_user_id	1-X	Max 6 numbers	N	
Donations	donation_id	1-X	Max 8 numbers, Unique and Auto Increment	N
	startdate	date	DATE, when the donation is created (CURRENT TIMESTAMP as default)	N
	amount	1-X	Max 8 numbers before comma, and Max 2 numbers after comma	N
	users_user_id	1-X	Max 6 numbers	N
	projects_project_id	1-X	Max 6 numbers	N
Support	ticket_id	1-X	Max 6 numbers, Unique and Auto Increment	N
	question	All Characters	Max 255 characters	A/N
	question_date	date	DATE, when the question is created (CURRENT TIMESTAMP as default)	N
	answer	All Characters	Max 255 characters	A/N
	answer_date	date	DATE, when the question is answered (CURRENT TIMESTAMP as default)	N
	users_user_id	1-X	Max 6 numbers	N
Ranking	rank_id	1-X	Unique and Auto Increment	N
	ranking_name	1-X	Either 1, 2, 3, 4 or 5	N
	projects_project_id	1-X	Max 6 numbers	N
Media	media_id	1-X	Auto Increment	N
	media_url	a-Å	Max 200 characters	A/N
	projects_project_id	1-X	Max 6 numbers	N
	media_type	img / Video	Eiter img or video	A/N
	date	date	DATE, when the media is created (CURRENT TIMESTAMP as default)	N
	primary_media	a-Å	Either y or n	A/N
Categories	category_id	1-X	Max 6 characters and Auto Increment	N
	category_name	a-Å	Max 45 characters	A/N