



Rainbow warriors gruppe:
Monika, Carina, Martin og Khava

Ansvarlige undervisere:
Luise Steenholdt, Niels Müller Larsen

Problemformulering	2
Metoder og process	2
User research og design	2
Persona	3
User journey	4
Design	4
Prototypen	7
Programmering	7
Programmering med Live Share og GitHub	7
APIs	7
Javascript DOM	8
PhP	8
User testing	8
Konklusion	9
Litteraturliste	9

Problemformulering

Til vores 1. semester eksamensprojekt har vi fået en problemstilling fra en repræsentant fra Tønder Festival, som har et ønske om en ny, brugervenlig web-app, der skal hjælpe pladskontoret med at varetage meddelelser om fejl m.m. under festivalen. Derfor har vi lavet en problemformulering, der lyder således:

Hvordan kan vi lette presset af alle henvendelser vedr. opståede problemer, der kommer ind til pladskontoret, samt strukturere de indkomne data bedst, sådan at fejlrapporter hurtigere og mere effektivt kan besvares og løses?

Metoder og process

For at nå vores mål, har vi først og fremmest benyttet Google Drive og projektværktøjet Trello for at skabe et overblik over alle de dele, vi vil have vores app til at gøre, og hvilke forskellige opgaver, der skal udføres, for at det lykkes. Vi har brugt metoder og teknikker, som vi har lært i undervisningen i både webprogrammering og brugeroplevelser, samt inddraget andre færdigheder inden for lignende emner fra hinanden.

Undervejs har vi haft en masse online møder over Microsoft Teams, hvor vi har snakket om og løbende fordelt alle opgaverne, som vores projekt involverer. Det har været en interessant og meget anderledes måde at kommunikere og arbejde på, men noget som vi sagtens har kunnet håndtere. Nogle gange er vi selvfølgelig også mødt op på skolen for at snakke sammen, når vi har haft store problemer med specifikke opgaver.

User research og design

Persona

Efter en formiddag med en masse brainstorming og agil planlægning, er vi blevet enige om at lære vores kommende brugere og deres behov lidt bedre at kende som det første. Vi er startet ud med at undersøge, hvilke muligheder, der er for at få et godt indblik i, hvordan frivillige hos Tønder Festival opfatter det egentlige problem - at den måde, Tønder Festivals fejlmelding-system fungerer på lige nu er alt for overskuelig og tager for lang tid, hvilket gør, at alle problemer måske ikke altid bliver løst.

Igennem en af Googles mange funktioner har vi derfor lavet et spørgeskema med nogle få, men grundige og meget specifikke spørgsmål, som vi har fået lov til at lægge ud på en Facebook-gruppe for Tønder Festival, hvor vi så har efterspurgt de frivillige arbejdere. Derudover har vi også haft kontakt til repræsentanten for Tønder Festival, Esben, som også har sendt vores spørgeskema ud til hans kollegaer, m.m.

Spørgeskemaet har vi løbende holdt øje med i nogle dage, og derefter har vi samlet de informationer, vi har modtaget, i en persona. Det har vi gjort for at skabe et overblik over vores målgruppe, da det ofte er lidt nemmere at arbejde med, når vi har noget konkret og visuelt foran os.



Gertrud, 34 år

Arbejde: Kontorassistent

Medier: Er stort set altid online - især på Twitter, fordi hun foretrækker den lette tone. Elsker at se eller lytte til god musik.

Udfordringer: Får ikke altid nok søvn. Arbejdet kan være hektisk. Kan ikke bare forlade baren, så hvis der opstår et problem, må man vente, til man kan få fat i en anden medarbejder på festivalen.

Status: Gift og har to børn.

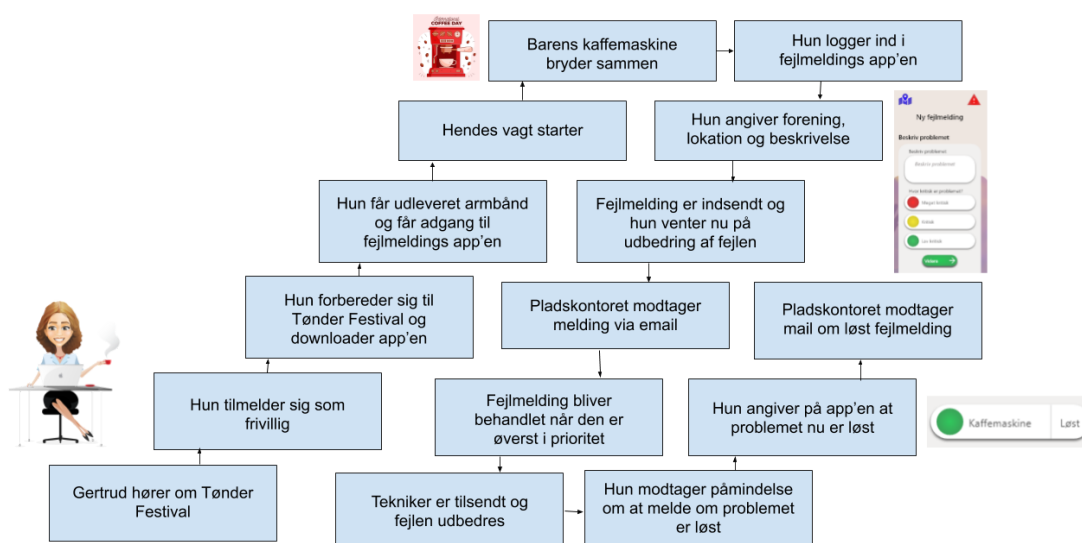
Beskrivelse: Gertrud er frivillig arbejder på Tønder festival. Hjælper med bar.

Rainbow warriors gruppe:
Monika, Carina, Martin og Khava

Ansvarlige undervisere:
Luise Steenholdt, Niels Müller Larsen

User journey

For at rigtig sætte os ind i, hvordan en frivillig medarbejder som Gertrud oplever en typisk arbejdssituation på Tønder Festival, har vi derudover skabt en oversigt over en af hendes dage som frivillig i form af en user journey.



Nu hvor vi kender vores målgruppe og dermed de frivillige hos Tønder Festival meget bedre, kan vi fortsætte vores arbejdsprocess - det næste, vi har valgt at tage hul på, er at planlægge design og funktioner af vores app. Det giver god mening, idet vi også har fået lidt design ideer m.m. ud fra svarene i vores spørgeskema.

Design

Vores designprocess involverer i første omgang vores eget moodboard flettet sammen med designmanualen, der er tilgængelig for os via Esben. Vi er startet med et moodboard for at få en fornemmelse af, hvilken stemning vi gerne vil fremkalde igennem app'en. Her har vi taget stor inspiration fra Tønder Festivals hjemmeside og ikke mindst deres mest benyttede farver og ikoner, som vi også har fundet i designmanualen.

Vi har oprettet to dokumenter til at klargøre, hvad for noget indhold, som vores app skal have. Vi har lavet et dokument, som vi kalder for functional requirements og så et dokument, der hedder content requirements. Til functional requirements skriver vi så i stikord en oversigt over, hvilke funktioner der skal være i vores app. F.eks. at der skal komme en alert, hvis man ikke har udfyldt et felt i formularen. I vores content requirement dokument har vi alt

det indhold, som skal være på vores app. Man kan kalde det for den informative del af indholdet. Her klargør vi hvilke informationer og indhold, som brugeren vil støde på. Det kan være fra den tekst, der forklarer ens fejlmeldinger eller informationen om, at brugeren skal slå notifikationer til. De dokumenter brugte vi så, da vi herefter lavede et flowchart over, hvor de forskellige funktioner og indhold skulle ligge.

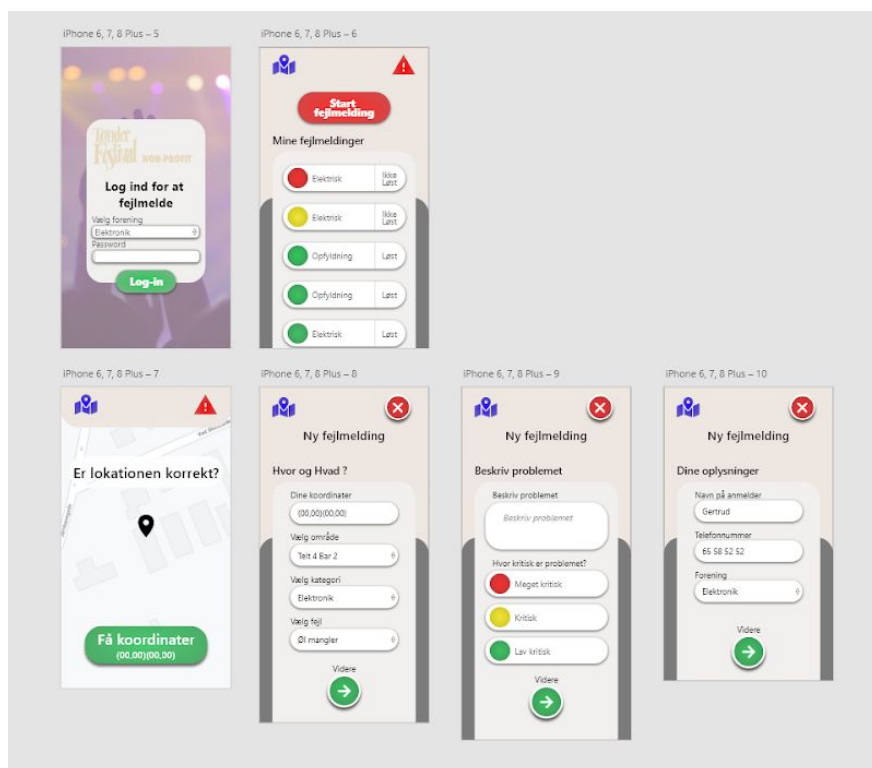
Ud fra alt dette har vi derefter skitseret forskellige wireframes med ideer til design af vores app. Vi har i første omgang gjort det traditionelt, da det er en af de mest effektive måder at få noget visuelt på hurtige, "rå" ideer. Vi har vist hinanden, hvad vi har tegnet, og derefter lavet to prototyper i Adobe XD, for til sidst at flette elementer sammen fra de to prototyper og skabe vores endelige mål for designet. Prototypen, som er præsenteret herunder via et screenshot fra Adobe XD, fokuserer på brugervenlighed med et simpelt, stilrent design, med en behagelige farveprofil, som er inspireret af Tønder Festivals nuværende identitet. Ydermere har vi valgt farverne rød, grøn og gul til at symbolisere niveauet af situationen, som skal fejlmeldes. Rød symboliserer noget, som er meget kritisk og grønt er mindre kritisk. De røde og grønne farver er også ført videre til de forskellige knapper, som vi bruger. Her er rød f.eks. brugt, når man skal starte en fejlmelding eller afslutte undervejs. Rød her symboliserer nogle mere kritiske valg. Videre knapper og indsend er grønne, da det er den naturlige vej at gå, når man er i gang inde i programmet. Log-ud knappen er også grøn. Det har vi taget fra virkeligheden. De fleste exit skilte er nemlig grønne.



Moodboard

Rainbow warriors gruppe:
Monika, Carina, Martin og Khava

Ansvarlige undervisere:
Luise Steenholdt, Niels Müller Larsen



Prototypen

Til ikonerne har vi benyttet os af metaphorical design, hvor et velkendt symbol repræsenterer en funktion, som man forventer pga. at alle brugere har set symbolet med samme funktion i mange andre situationer. Det gør vi for at gøre det nemmere for brugerne at vide, hvor de skal trykke, så de får den bedst mulige oplevelse af at bruge vores app.

Programmering

Programmering med Live Share og GitHub

Efter udviklingen af prototypen har vi ved hjælp af pseudo-kode lagt vores plan til, hvad app'en skal indeholde, og er klar til at begynde at programmere. Til dette bruger vi alle Visual Studio Code og har for første gang benyttet en plugin kaldet Live Share, som gør, at vi kan se hinanden arbejde i vores kode live. Selvom det er lidt af en udfordring at finde ud af, så har vi lært det relativt hurtigt. Det er et godt lille værktøj, da vi finder det meget nemmere at programmere et fælles projekt, når vi hver især kan se, hvad vi alle laver live. På den måde kan vi nemmere flette vores koder sammen og sikre os, at alt fungerer

Rainbow warriors gruppe:
Monika, Carina, Martin og Khava

Ansvarlige undervisere:
Luise Steenholdt, Niels Müller Larsen

sammen, i modsætning til de problemer, der kan opstå, hvis man splejser forskelligt kode sammen - det kan især blive problematisk, hvis man hver især har meget forskellige måder at kode på.

Derudover er GitHub også en af vores primære værktøjer til at holde styr på vores projekt, da det er det, vi er vant til at bruge, og det sikrer os, at vi hele tiden har en opdateret version af vores projekt tilgængeligt for alle gruppemedlemmer til enhver tid, da det fungerer lidt som et online drev kun til vores projekt - i dette tilfælde.

APIs

Vi har benyttet os af HTML5 Geolocation Api som er indbygget i HTML5. Dette API er tilgængeligt for alle browsere som kører HTML5 og er givet med objektet *navigator.geolocation*. For at hente enhedens position kan man bruge metoden *getCurrentPosition()*. Det kræver brugerens accept at bruge enhedens placering og der vil automatisk komme en prompt til at give applikationen lov. Det skal siges at når man har med følsom data at gøre bør man bruge den krypterede *https* protokol for at sikre sig mod at brugeren kan spores. Vi har også benyttet os af timestamp som også kan fås fra *navigator.geolocation* objektet. Det er givet i millisekunder siden Unix epoken og vi har derfor formateret det efter lokal tidsregning med metoden *Date()*.

Det andet API vi har brugt er hentet eksternt fra MapQuest som udbyder kort funktionalitet. Det er ganske gratis indenfor en vis mængde forespørgsler. Da brugeren kun skal verificere sin lokation har vi valgt at bruge et statisk kort. Det fungerer ved at man angiver parametre i et link samt sit API key og tilbage får man et billede i .png format.

Javascript DOM

Vi har brugt DOM mange steder i vores kode. Funktionen til vores fejloversigt bruger både *createElement* og *setAttribute* i en for-løkke, sådan at den laver elementer for hver fejl i det array, vi har brugt til oversigten. DOM viste sig at være en stor hjælp da vi skulle sende data fra bruger inputs videre til vores mail handler. Da PHP tager fat i *values* fra elementer som er givet ved *name* attributten kunne vi bruge *createElement* og *setAttributeNode* til at angive at PHP'en skal tage values fra vores niveau knapper, når de er klikket.

PhP

PhP ligner meget JavaScript i sin opbygning. PhP er et server sprog. Dette betyder, at al kode og data i PhP bliver kørt på serveren, som herefter sender resultater tilbage som HTML til brugerens browser.

Vi er startet med at skabe et output i PhP, altså noget som browseren vil vise. Det fungerer ved at serveren kigger dit PhP dokument igennem, udfører den kode man har skrevet, og returnerer formular. Det er noget brugeren kan modtage pr mail, som i vores tilfælde er informationer vedr. fejlmeldinger fra brugeren af vores app.

User testing

Når vi er nået det punkt, hvor vores app fungerer, skal vi selvfølgelig afprøve det hele over for nogle andre end os selv i form af en user test. Vi har tænkt over, hvad for en slags user test, vi vil lave, og hvem, vi vil række ud efter for at hjælpe os med det. Efter at vi har gennemgået et par forskellige muligheder, har vi valgt at lave en tænke-højt-test meget ligesom den, vi har lavet i klassen i forbindelse med projektet i brugeroplevelser. Det er nemlig en effektiv og hurtig metode til bl.a. at aflæse på en brugers reaktioner, om der er noget, der skal laves om - eller om der er noget i vores design, som fungerer rigtig godt.

Vi har tænkt os at udvælge en person, som er frivillig hos Tønder Festival til denne opgave, og så har vi derudover overvejet et par andre frivillige deltagere, som selvfølgelig vil sætte sig ind i rollen som en medarbejder for Tønder Festival. Vi vil derefter give dem en lille række opgaver, som de skal udføre, imens vi instruerer dem i at sige alt, hvad de umiddelbart tænker om vores app - både om designet og funktionaliteten, uanset om det er godt eller skidt.

På grund af vores planlægning af tid har vi dog ikke haft muligheden for at følge vildt meget op på planen med vores user test, men det er jo altid godt at have en plan.

Konklusion

I slutningen af projektperioden står vi nu klar med en app, som visuelt repræsenterer det meste af det, vi originalt havde tænkt os. Funktionerne har også fulgt vores ide mht. prototypen ret godt, men der har dog været noget problematik undervejs, som har gjort, at

Rainbow warriors gruppe:
Monika, Carina, Martin og Khava

Ansvarlige undervisere:
Luise Steenholdt, Niels Müller Larsen

der har været funktioner, vi har måtte ændre - fx er vores kort ikke helt som originalt planlagt.

Med en stilren profil, som følger Tønder Festivals farve-identitet og har få, men genkendelige knapper, har vi opnået at skabe en nem og overskuelig løsning for de tiltænkte brugere af vores app. Den er simpel og ligetil, og sender de vigtigste informationer vedr. fejlmeldinger gennem en nem og kort process fra brugerens side. Derfor mener vi, at vi har opnået et fint resultat i forhold til vores problemformulering.

Litteraturliste

“CSS Tutorial”. Set 16. december 2020. <https://www.w3schools.com/css/default.asp>.

“How to Find Position with HTML5 Geolocation - Tutorial Republic”. Set 16. december 2020. <https://www.tutorialrepublic.com/html-tutorial/html5-geolocation.php>.

“Hvad er PHP?” Set 14. december 2020.
<https://www.nemprogrammering.dk/Tutorials/Startviden/teknologierne/php.php>.

w3resource. “JavaScript : HTML form - Checking for non empty”. Set 16. december 2020. <https://www.w3resource.com/javascript/form/non-empty-field.php>.

Stack Overflow. “jquery - I want to store Javascript array as a Cookie”. Set 16. december 2020.
<https://stackoverflow.com/questions/2980143/i-want-to-store-javascript-array-as-a-cookie>.

MapQuest API Documentation. “Static Map API - Getting Started”. Set 16. december 2020.
<https://developer.mapquest.com/documentation/static-map-api/v5/getting-started/>.

“Web Development”. Set 16. december 2020. <http://dkexit.eu/webdev/site/index.html>.