

Bachelor

Bachelor

*Bachelor in Software Development,
IT-University of Copenhagen*

Jakob Melnyk, jmel@itu.dk
Frederik Lysgaard, frly@itu.dk

May 22st, 2012

Indholdsfortegnelse

1	Baggrund for projektet	2
2	Design af brugergrænsefladen	6
3	Usability testing	12

1 Baggrund for projektet

1.1 Udfordringen ved et bookingsystem

I lang tid har lokale booking på ITU været et uoverskueligt og problemfyldt område og har forhindret at ITU's lokaler kunne bruges optimalt. Vi vurderer derfor at det er vigtigt at udvikle et ordentligt lokale bookingsystem således at ITU kan udnytte deres mange lokaler optimalt. Det udfordrende ved et bookingsystem er, at det er et servicesystem der både skal understøtte services til booking af lokaler og opdatering og vedligeholdelse af systemet. Der er derfor mange aktører der skal tages hensyn til, og mange arbejdsområder. For at have overblik over hvad der skulle understøttes og hvad der var vigtigst, fulgte vi en kravspecifikation som blev udarbejdet i kurset "Anskaffelse og kravspecifikation" på ITU i efteråret 2012. Kravspecifikationen er lavet med fokus specifikt på ITU og hvad et bookingsystem til ITU kræver.

1.2 Kravspecifikationen og ændringer

I kravspecifikationen beskrives hvordan den nuværende booking situation er på ITU, i øjeblikket foregår bookingen af lokaler i mange forskellige systemer, hvor der er en aktør som fungerer som bindeled for administrationen af lokaler. For at forbedre den løsning foreslår kravspecifikationen at der laves et system som alle aktørerne arbejder op imod. Det er så vidt muligt den løsning vi vil lave en begrænset implementering af. Figur ?? viser den nuværende situation.

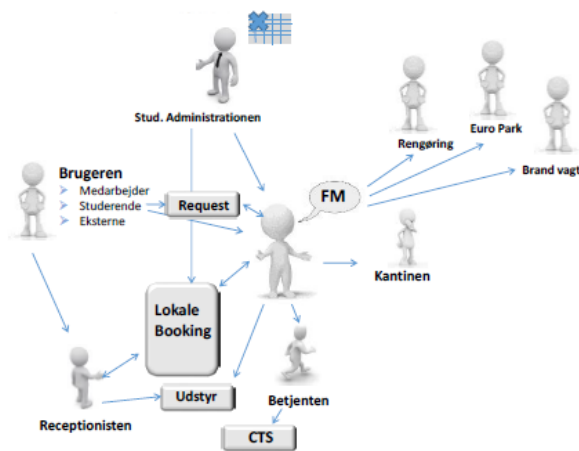


Figure 1.1: Nuværende system flow for booking systemet på ITU

1.2.1 Ændringer til kravspecifikationen

Kravspecifikationen præsenterer en datamodel, der beskriver hvilke informationer systemet skal bruge. Vi har lavet nogle ændringer så den passer til vores implementering af bookingsystemet. Vi har fjernet lokale status og lavet en "mange til én" forbindelse mellem booking og lokale da vi gerne ville have at en booking kun kunne have et lokale. Vi fjernede også "Lokale egenskaber" da vi vurderede, at det gav mest mening hvis "Inventar" repræsenterede et unikt stykke inventar og ikke havde antal i "Lokale egenskaber". Vi slettede også alle "pris" værdier fra modellen, da vi ikke havde nogen intention om at understøtte fakturering.

1.3 Arbejdsopgaver

Denne sektion beskriver hvilke arbejdsopgaver vi understøtter i første release samt giver en prioritering af, hvilke arbejdsopgaver der bør fokuseres på i følgende releases.

1.3.1 Arbejdsområde 1: Booking

C1: Administrer booking

Arbejdsopgave	
C1.1	Understøttet
C1.1a	Understøttet
C1.2	Understøttet
C1.2a	Understøttet
C1.3	Understøttet
C1.4	Understøttet
C1.5	Understøttet

C2: Forespørg på booking

Arbejdsopgave	
C2.1	Understøttet
C2.1a	Understøttet
C2.2	Understøttet
C2.3	Understøttet

1.3.2 Arbejdsområde 2: Booking behandling

C3: Modtag forespørgsel fra Eksern

Arbejdsopgave	
C3.1	Understøttet
C3.2	Understøttet
C3.2a	Understøttet
C3.3	Ikke understøttet
C3.4	Understøttet
C3.5	Understøttet
C3.6	Ikke understøttet

C4: Klargør lokaler

Arbejdsopgave	
C4.1	Ikke understøttet
C4.2	Ikke understøttet
C4.3	Ikke understøttet

C5: Udlever nogle

Arbejdsopgave	
C5.1	Ikke understøttet
C5.2	Ikke understøttet

1.3.3 Arbejdsområde 3: Forplejning

C6: Håndter forplejning

Arbejdesopgave	
C6.1	Ikke understøttet
C6.2	Ikke understøttet
C6.3	Ikke understøttet
C6.4	Ikke understøttet

C7: Fakturer forplejning

Arbejdesopgave	
C7.1	Ikke understøttet
C7.5	Ikke understøttet

C8: Opadter menukort

Arbejdesopgave	
C8.1	Ikke understøttet

1.3.4 Arbejdsområde 3: Systemadministration

C9: Opdater listen for ekstra udstyr

Arbejdesopgave	
C9.1	Understøttet

C10: Opdater lokaler

Arbejdesopgave	
C10.1	Understøttet
C10.2	Understøttet

C11: Administrer bruger

Arbejdesopgave	
C11.1	Ikke understøttet
C11.1a	Ikke understøttet

C12: Håndter statistik

Arbejdesopgave	
C12.1	Ikke understøttet
C12.2	Ikke understøttet

1.3.5 Arbejdsområde 4: Finans

C13: Behandl faktura

Arbejdsopgave	
C13.1	Ikke understøttet
C13.2	Ikke understøttet

1.3.6 Prioritering af arbejdsopgaver

Til den første release af systemet valgte vi at fokusere på at understøtte systemadministration og booking af lokaler og udstyr, hvilket tydeligt kan ses i afsnit ?? da alle arbejdsopgaver i arbejdsområderne C1,C2,C3,C9 og C10 er understøttet. I kravspecifikationen er de arbejdsområder også blandt dem som er vægtet højest og det er samtidig de arbejdsområder som indeholder kernefunktionerne i systemet. Det som ikke blev prioriteret højt nok til at komme med i første release var de arbejdsopgaver som fokuserede på integrationen med kantinen, arbejdsopgaverne C6.1- C6.4 er eksempler på sådanne opgaver. Grunden til at vi ikke tog de arbejdsopgaver med var, at vi vurderede, at de ikke var nødvendige for at have et system der understøttede basis funktioner, derudover lagde arbejdsopgaverne op til, at der skulle implementeres et interface specifikt til kantinen hvilket vi vurderede ville tage lang tid at implementere og vi ville også have tre brugergrupper at tage hensyn til ift. usability og testing. Fokus for næste release vil derfor være at få udarbejdet et interface som kantinen kan bruge til at integrere med resten af systemet og få finpudset de allerede eksisterende funktioner, i kravspecifikationen. Behandling af faktura har også en høj vægtning så det vil der også være fokus på.

2 Design af brugergrænsefladen

Dette kapitel beskriver det generelle design af brugergrænsefladen samt de beslutninger, som ligger bag.

2.1 Generelle mål

Vi har valgt at designe vores brugergrænseflade ud fra reglerne om design af virtuelle vinduer[SL, s. 169] samt Ease Of Use principperne[SL, s. 9]. I forbindelse med dette valg har vi sat følgende mål for designet:

- Strømlinet brugergrænseflade
- Få forskellige skærbilleder
- Overblik
- Effektivt

2.1.1 Strømlinet brugergrænseflade

Vi har valgt at designe skærbillederne med samme grundstruktur. Denne lighed bør gøre det intuitivt at gå fra et skærbillede til et andet i forbindelse med udførsel af opgaver. Desuden følger det designregel 1¹ om få vindueskabeloner.

2.1.2 Kort vej fra en opgave til en anden

Brugergrænsefladen skal gøre det hurtigt og nemt for brugeren at komme fra en opgave til en anden. Dette skal gøres ved at have få skærbilleder involveret i en enkelt task (designregel 2²).

2.1.3 Overblik

Brugeren skal have mulighed for nemt at danne sig overblik over bookinger, udstyr og forplejning (regel 6³). Derfor skal vi have separate skærbilleder, som giver overblik over hver type.

2.1.4 Effektivt

Det skal være effektivt at udføre opgaver for brugere, som anvender systemet ofte.

2.2 Brugergrænsefladens udvikling og udseende

Vores skærbilleder er opdelt i tre typer: Gitter, Almindelig og Pop-ups.

Gitterskærbillederne bruger vi til booking af lokale, forplejning og udstyr samt administration af udstyr og lokaleinventar.

Almindelige vinduer anvender vi, hvis man skal ændre noget på et stykke udstyr/inventar eller et lokale.

Pop-up skærbilleder er generelt advarsler eller fejlbeskeder.

¹Few window templates

²Few window instances per task.

³Necessary overview of data

2.2.1 Gitterskærmbilledet

Gitterskærmbillederne er de skærmbilleder som står for at visse hvad der kan bookes i vores system, lokaler, udstyr og forplejning. Figur 2.1, side 7 og Figur2.2, side 7 er screenshots af vores første udkaste til gitterbilledet og vores endelige design.

The screenshot shows a web interface titled "LOKALELISTE". At the top, there is a legend: a red square for "= Booket", a white square for "= Ledig", and a blue square for "= Dine bookinger". Below the legend is a "Dato:" field. The main part of the interface is a table with columns for "Navn", "Kapacitet", "Udstyr", and time slots from "09:00" to "20:00" in one-hour increments. The table contains several rows, each representing a room, with checkboxes in the time slot columns for booking.

Figure 2.1: Den endelige udgave af gitteret

The screenshot shows a hand-drawn mockup of the booking grid. It includes a header with "Lokale", "Kapacitet", "Udstyr", and "Dine Bookinger". Below this is a table with columns for "Lokale", "Capacity", "Equipment", and time slots from "9" to "21". The table contains three rows of data: "Audi 2" with capacity 124 and equipment "pro., kanna, oply.", "2a12" with capacity 30 and equipment "projektor", and "2a01" with capacity 20 and equipment "- - -". The mockup also includes a "Dato:" field set to "22/03-2013", a "Side 1 2 3 4 5 >" navigation bar, and a "Book" button.

Figure 2.2: Første udgave af gitter layoutet

Vores første mockup af gitterskærmbilledet havde et gitter, hvor hver række var et lokale og tiderne var kolonner. Brugere skulle klikke i et felt for at vise, at man ønskede at booke på et bestemt tidspunkt. Når man havde valgt de tider og lokaler, man gerne ville booke, skulle man trykke på en "Book" knap. Som man kan se på Figur 2.1 valgte vi at beholde den struktur, men vi tilføjede checkboxes til gitteret. Dette blev gjort efter første rund af usability tests hvor vi observerede, at brugeren var i tvivl om hvor man skulle klikke for at vælge tider for en booking, derudover tilføjede vi også, at udstyr m.m., man har booket, ligger

2.2. BRUGERGRÆNSEFLADENS UDVIKLING OG UDSEENDE

øverst i gitteret, når man får oversigten over fx udstyr. Dette gør det nemt at finde det element, man har booket/bestilt.

2.2.2 Almindelige skærbilleder

Disse typer af skærbilleder er primært til administration af udstyr/inventar/lokale og lignende. Et eksempel er skærbilledet til ændring af et stykke udstyr (Figur 2.3, side 8).

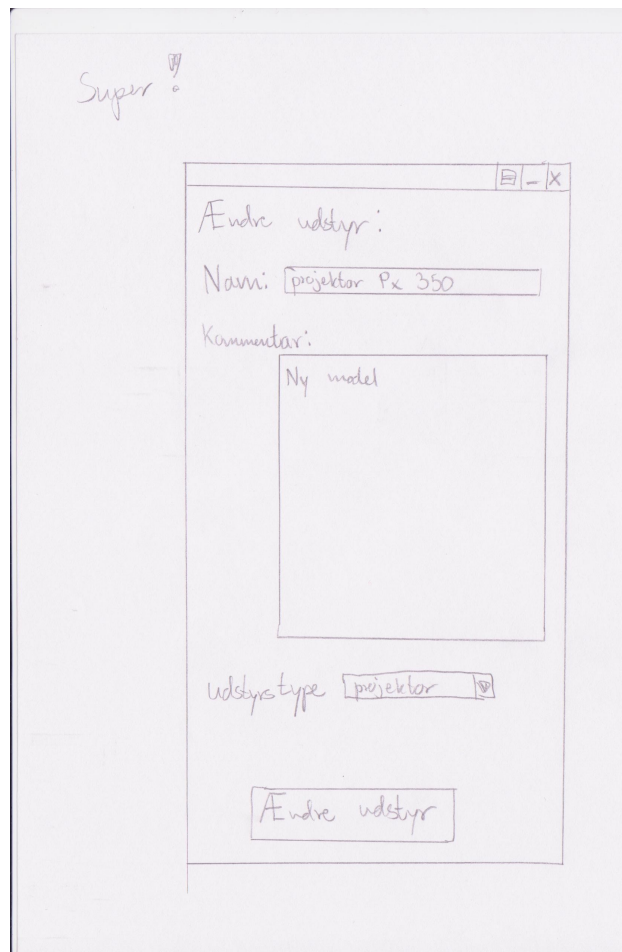


Figure 2.3: Papermockup af skærbilledet til ændring af udstyr

Ændre udstyr

Som man kan se på Figur 2.3 var det første skærbillede til Ændreudstyr smalt i forhold til det endelig, Figur 2.4. Grunden til det blev ændret skyldtes at vi fulgte designregell og derfor gerne ville have at billederne var visuelt konsistente.

Dine bookinger

Som man kan se på Figurerne 2.5 og 2.6 var den eneste ændringer at vi fjernede forklaringen til hvad de forskellige statusser repræsenterede. Vi har i designet af skærbilledet 2.6 fokuset på at overholde Gestalt-

2.2. BRUGERGRÆNSEFLADENS UDVIKLING OG UDSEENDE

BookIT

[Book lokale](#) [Udstyrsliste](#) [Booking-liste](#) [Dine bookinger](#) [Lokaleliste](#) [Find bookinger](#)

ÆNDRE Udstyr

Name:

Kommentar:

☐ Inventar

☐ Udstyr

Udstyrstype

Projektor

Ændre udstyr

Figure 2.4: Skærbilledet til ændring af udstyr

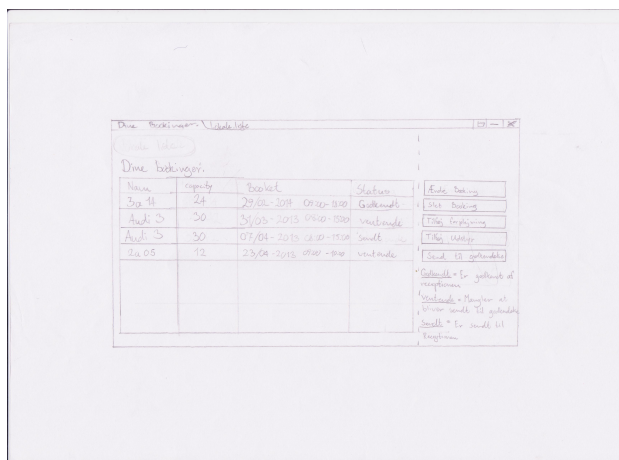


Figure 2.5: Papermockup af skærbilledet til visning af bookinger

BookIT

[Book lokale](#) [Udstyrsliste](#) [Booking-liste](#) [Dine bookinger](#) [Lokaleliste](#) [Find bookinger](#)

DINE BOOKINGER

Lokale	Kapacitet	Booket	Status	Forplejning	Udstyr
ZA12	30	09:00-12:00	Godkendt		Projektor
3A14	30	09:00-12:00	Godkendt		Projektor
Aud2	30	09:00-12:00	Godkendt		Projektor

Ændre booking

Slet booking

Tilføj forplejning

Tilføj Udstyr

Send til godkendelse

Figure 2.6: Skærbilledet til visning af bookinger

lovene[SL, s. 68], specifikt Law of Proximity⁴, så det virkede naturligt at knapperne til højre hørte til gitteret.

Godkend bookinger

Figurerne 2.7 og 2.8 viser henholdsvis papermockupen og det færdige skærbilledede der viser listen over hvilke bookinger der mangler at blive godkendt eller afvist af receptionisten. Forskellen fra papermockupen til det færdige skærbillede er, at vi har valgt at flytte godkend og afvis knapperne fra at være tilknyttet til hver enkelt booking til at være uden for gitteret, således at vi ligsom Figur 2.6 benytter "Law of proximity".

⁴"Pieces that are close together are perceived as belonging together".

2.2. BRUGERGRÆNSEFLADENS UDVIKLING OG UDSEENDE

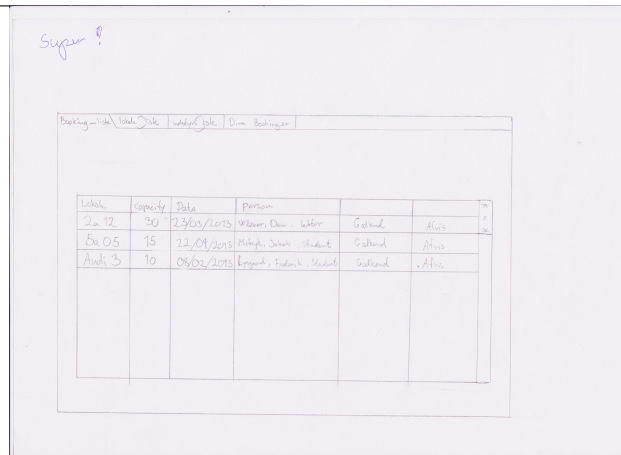


Figure 2.7: Papermockup af skærbilledet til visning af bruger bookinger

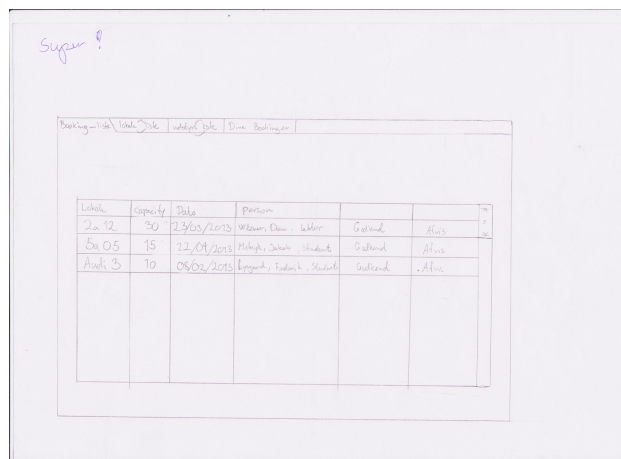


Figure 2.8: Skærbilledet til visning af bruger bookinger

Udstyrsliste

Figurerne 2.9 og 2.10 viser papermockupen og det endelige skærbillede af listen over udstyr og tilføjelse af nyt udstyr til systemet. Til designet af skærbilledet har vi brugt "Law of proximity" så det er tydeligt at de to funktioner i skærbilledet ikke hører sammen.

2.2. BRUGERGRÆNSEFLADENS UDVIKLING OG UDSEENDE

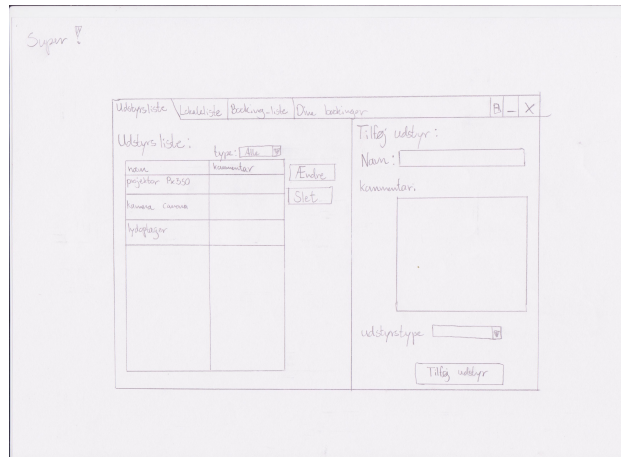


Figure 2.9: Papermockup af skærbilledet til tilføjelse og visning af udstyr

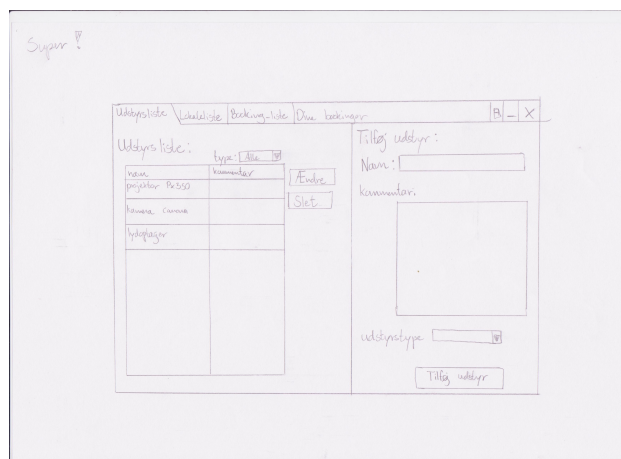


Figure 2.10: Skærbilledet til tilføjelse og visning af udstyr

3 Usability testing

Dette kapitel beskriver, hvad vores strategi for usability testing har været samt resultaterne af vores tests.

3.1 Test strategi

Kravspecifikationen beskriver, at det er nødvendigt at lave usability tests tidligt i processen for at bevise proof of concept. Vi lavede papermockups af vores skærbilleder for at opfylde dette krav.

Vi valgte at lave to runder af usability tests.

Runde 1 bestod af tests med en papermockup på en enkelt bruger.

Runde 2 bestod af tests af det endelige produkt på flere brugere.

Vi valgte kun at teste én bruger i runde 1, da flere testpersoner tidligt i udviklingsfasen kan føre til en overvældende liste af problemer[SL, s. 416]. Derudover er en enkelt testperson ofte nok til at afsløre de mest seriøse problemer i brugergrænsefladen.

Vi mener, at det havde været optimalt, hvis vi kunne have udført en usability test midtvejs i processen (mellem runde 1 og runde 2), men vi vurderede, at vi ikke ville få nok ud af testen, da det ville være svært at finde tid til at teste nok personer samt implementere de mulige ændringsforslag.

3.1.1 Test cases

Vi har to brugergrupper i vores usability tests. De er almindelige brugere (studerende/undervisere) og administration (Facility Management/Receptionister). Til hver af disse brugergrupper har vi defineret syv test cases.

Test cases til almindelige brugere:

Test Case 1: Du har booket et lokale fra 9-10 til et møde med din vejleder og du vil gerne have en diktafon med, så du kan optage mødet.

Test Case 2: Du skal holde et møde på tre timer om et par dage. Du skal i den sammenhæng booke et lokale til formålet.

Test Case 3: Et af dine gruppemedlemmer har meldt tilbage, at han skal aflevere sin datter i børnehaven. Mødet kan derfor først holdes fra klokken 10.

Test Case 4: Da mødet ligger om morgen, tænker du, at det vil være fornuftigt hvis der var noget kaffe og morgenbrød klar.

Test Case 5: Du skal til dit projekt optage en kort reklamefilm og har derfor brug for et kamera.

Test Case 6: Et af dine gruppemedlemmer har kamera med, så du behøver ikke længere booke et hos ITU.

Test Case 7: Deltagerne til dit møde i 2a12 har desværre aflyst.

Test cases til administrationsbrugere:

Test Case 1: En elev har kontaktet dig og gjort dig opmærksom på, at der ikke længere er en projektor i 2A12.

3.2. RUNDE 1: RESULTATER OG KONSEKVENSER

Test Case 2: ITU har til sin udstyrssamling erhvervet sig to nye kameraer.

Test Case 3: En student henvender sig til dig ved skranken og spørger om hans booking er blevet godkendt.

Test Case 4: Du har et møde den i de kommende dage og vil helst gerne holde det på 4. sal.

Test Case 5: Du skal afholde et foredrag for 30 mennesker den 24, tidspunktet er i relevant men der skal være plads til min. 30 mennesker.

Test Case 6: Lokalet 5a12 er ikke længere et privat lokale, men skal istedet registreres som et mødelokale.

Test Case 7: Lokalet 2a12 er ikke længere til rådighed i forbindelse med booking.

Vi har designet vores test cases således, at testpersonen ikke får at vide hvor i systemet, de skal udføre deres arbejdsopgave. Dette har vi gjort for at finde ud af, hvor intuitivt vores system er.

De 14 test dækker til sammen størstedelen af systemets workflows. Vi udfører testene som "think aloud" tests [SL, s. 421]. "Think-aloud" tests består af at læse test casen op for testpersonen og bede dem tænke højt og forklare, hvad de gør under testen.

Efter testen får brugeren mulighed for at give input til, hvad de synes er godt og hvad der kan forbedres. Dette giver os mulighed for at få et indtryk af, hvad vi har overset i vores design og hvilke elementer, vi bør bruge flere steder i systemet.

I første runde af tests havde vi fokus på de store problemer i systemet, hvor vi i anden runde havde fokus på at følge op på de ændringer, vi havde lavet i forhold til resultaterne fra første runde. I anden runde var vi også interesserede i at finde ud af, om vores system er intuitivt og effektivt nok.

3.2 Runde 1: Resultater og konsekvenser

Efter den første runde af tests var der fem problemer, problemerne er klassificeret i forhold til skallen [SL, s. 439]:

Punkt 1: Minor problem - Usikker på hvordan man vælger booking tidspunkter.

Punkt 2: Cumbersome - Brugeren synes det var besværlig at navigere imellem skærbillederne.

Punkt 3: Medium problem - Svært ved at forstå sletning af forplejning/udstyr og booking.

Punkt 4: Minor problem - Problem med at tilføje udstyr/forplejning.

Punkt 5: Positiv feedback på den generelle struktur af systemet.

Punkt 1 Test personen havde problemer med at finde ud af, hvordan vores gitter til booking af lokaler fungerede. Personen troede ikke, man kunne klikke på felterne i gitteret, så der skulle lidt hjælp til, før det personen forstod princippet.

Konsekvenser og løsning Vi blev enige om, at dette problem kunne løses ved at indsætte en checkbox i felterne. Det vil gøre det mere intuitivt, når man skal trykke på tiderne.

Punkt 2 Testpersonen følte, at der manglede navigeringsmuligheder mellem de forskellige skærbilleder. Det var ikke intuitivt, at man skulle bruge tabsne til at skifte skærbillede.

Konsekvenser og løsning Vi løste problemet ved at lave en menubar, som var placeret nærmere midten af billedet, så den blev mere synlig.

3.3. RESULTATER AF USABILITYTEST RUNDE 2

Punkt 3 Testpersonen fandt det ikke naturligt, at man skulle bruge lokalelisten til at ændre sin booking.

Konsekvenser og løsning Da det var første test af vores system og det kun var teste på en person valgte vi ikke at gøre noget ved problemet da vi mente at vores grundstruktur i systemet var solid og derfor ikke burde ændres kun på grund af det problem.

Punkt 4 Testpersonen nævnte, at hun ikke mente, det burde være muligt at tilføje udstyr/forplejning til en booking før, den er blevet godkendt.

Konsekvenser og løsning Vi valgte at imødekomme dette ved at implementere logik, så man ikke kan trykke på knapperne til at tilføje forplejning/udstyr, hvis man ikke har valgt en godkendt booking.

Punkkt 5 Testpersonen kunne godt lide den overordnede struktur af systemet og var positiv overfor booking funktionaliteten (gitteret). Personen skulle bare lære at bruge booking funktionaliteten.

Konsekvenser og løsning Denne feedback havde stor indflydelse på, at vi valgte at beholde vores grundstruktur.

3.3 Resultater af usabilitytest runde 2

3.4 Mulige konsekvenser af usabilitytest runde 2

Bibliography

[SL] Lauesen, Søren. User Interface Design, A Software Engineering Perspective. Great Britain: Pearson Educated Limited, 2005. Print.