

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	5
3	Usability testing	9

1 Baggrund for projektet

1.1 Udfordringen ved et booking system

I lang tid har lokale bookning på ITU været et uoverskueligt og problemfyldt område og har forhindret at ITU lokale kunne bruges optimalt. Vi vurderer derfor at det er vigtigt at udvikle et ordenligt lokale booking system således at ITU kan udnytte deres mange lokaler optimalt. Det udfordrende ved et booking system 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 booking system til dem 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 aktørene arbejder op imod. Det er så vidt muligt den løsning vi vil lave en begrænset implementering af.

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 booking systemet. Vi har fjernet lokale status og lavet en mange til en 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

1.3. ARBEJDSOPGAVER

C2: Forespørg på booking

Arbejdesopgave	
C2.1	Understøttet
C2.1a	Understøttet
C2.2	Understøttet
C2.3	Understøttet

1.3.2 Arbejdsområde 2: Booking behandling

C3: Modtag forespørgsel fra Eksern

Arbejdesopgave	
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

Arbejdesopgave	
C4.1	Ikke understøttet
C4.2	Ikke understøttet
C4.3	Ikke understøttet

C5: Udlever nøgle

Arbejdesopgave	
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

1.3. ARBEJDSOPGAVER

C8: Opdater menukort

Arbejdsopgave	
C8.1	Ikke understøttet

1.3.4 Arbejdsområde 3: Systemadministration

C9: Opdater listen for ekstra udstyr

Arbejdsopgave	
C9.1	Understøttet

C10: Opdater lokaler

Arbejdsopgave	
C10.1	Understøttet
C10.2	Understøttet

C11: Administrer bruger

Arbejdsopgave	
C11.1	Ikke understøttet
C11.1a	Ikke understøttet

C12: Håndter statistik

Arbejdsopgave	
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 valgt vi at fokusere på at understøtte systemadministration og booking af lokaler og udstyr, hvilket tydeligt kan ses i afsnit 1.3 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 også de arbejdsområder som indholder corefunktionerne i systemet. Det som vi ikke priorerede i den første release var de arbejdsopgaver som fokuserede på integrationen med kantien, arbejdsopgaverne C6.1- C6.4 er eksempler på sådanne opgaver. Grunden til vi ikke tog de arbejdsopgaver med var, at vi vurderede, at de ikke var nødvendige for at have et system der understøttede basic funktioner, derudover lagde arbejdsopgaverne op til at der skulle implementeres et interface specifikt til kantien. Fokus for næste release vil derfor være at få udarbejdet et interface som kantien kan bruge til at integrere med resten af systemet og få finpudset de allerede eksisterende funktioner, i kravspecifikationen har behandling af faktura 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 Grundstruktur

Vores brugergrænseflade er delvist inspireret af booking systemer til biografer. Inspirationen består i, at vi laver en visning af lokalerne og får brugeren til grafisk at vælge lokale og tider for deres bookning. Målet var at gøre dette med farvekodning, således at brugeren nemt kunne skelne mellem ledige og optagede lokaler.

Vores første mockup havde et gitter, hvor hver række var et lokale og tiderne var kolonner. Brugerne skulle klikke i et felt for at vise, at man ønskede at booke på et bestemt tidspunkt. Når man havde valgt de tider

¹Few window templates

²Few window instances per task.

³Necessary overview of data

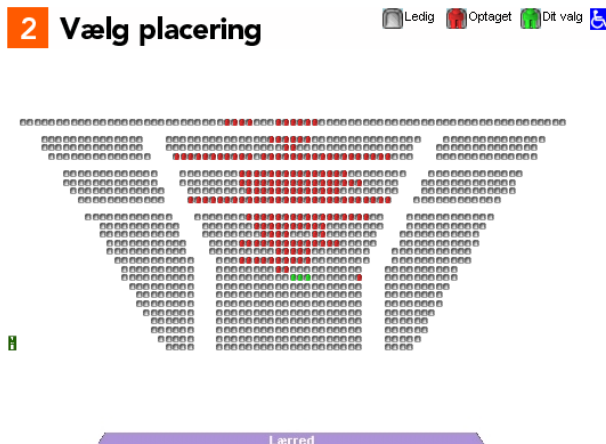


Figure 2.1: Screenshot af kino.dk's billet booking brugergrænseflade

og lokaler, man gerne ville booke, skulle man trykke på en "Book" knap.

Resultatet af vores første usability test (se 3.2, side 10) viste dog, at denne løsning ikke var intuitiv nok for brugeren. Vi valgte derfor at lave checkboxes inde i hvert felt, således at brugeren skal sætte et hak i de checkboxes, som brugeren vil booke tider ud fra.

Vi anvendte gitterløsningen (se figur 2.2, side 7) som grundstrukturen for de fleste skærbilleder for at følge vores mål om at lave en strømlinet brugergrænseflade.

Til resten af skærbillederne har vi holdt os til simple strukturer, som kun præsenterer den nødvendige data til at udføre arbejdsopgaven.

2.3 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.

2.3.1 Gitterskærbilledet

Figur 2.2, side 7 er et screenshot af vores endelige design af booking skærbilledet.

Vi tilføjede efter usability test 1, at udstyr m.m., man har booket, ligger øverst i gitteret, når man får oversigten over fx udstyr. Dette gør det nemt at finde det element, man har booket/bestilt.

Figur 2.3, side 7 er det papermockup, som blev brugt til første runde af usability tests. Forskellen på de to skærbilleder er de checkboxe, som er blevet tilføjet inde i gitteret.

2.3.2 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 2.4, side 7).

2.3. BRUGERGRÆNSEFLADENS UDVIKLING OG UDSEENDE

LOKALELISTE

☒ = Booket
 ☐ = Ledig
 ☒ = Dine bookinger

Dato:

Navn	Kapacitet	Udstyr	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2.2: Den endelige udgave af gitteret

Lokale: / Dine Bookinger:

☒ = Booket
 ☐ = ledig
 ☒ = dine

Lokaleliste:

Sal: Dato:

Lokale	Capacity	Equipment	9	10	11	12	13	14	15	16	17	18	19	20	21
Audi 2	124	pro., kanna, cykler osv													
2a12	30	projektor													
2a01	20	- - -													

Side 1 2 3 4 5 > >>

Book

Figure 2.3: Første udgave af gitter layoutet

BookIT

[Book lokale](#)
 [Udstyrliste](#)
 [Booking site](#)
 [Dine bookinger](#)
 [Lokaleliste](#)
 [Find bookinger](#)

ÆNDRE UDSTYR

Navn:

Kommentar:

Inventar: ☐ Udstyr: ☐

Udstyrstype:

Ændre udstyr

Figure 2.4: Skærbilledet til ændring af udstyr

2.3. BRUGERGRÆNSEFLADENS UDVIKLING OG UDSEENDE

BookIT					
Book lokale	Udstyrsliste	Booking-side	Dine bookinger	Lokaleliste	Find bookinger
DINE BOOKINGER					
Lokale	Kapacitet	Booket	Status	Forplejning	Udstyr
3A12	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 2.5: Skærbilledet til visning af bookinger

Derudover har vi i udviklingen af vores design gået efter at overholde Gestalt-lovene[SL, s. 68], specifikt Law of Proximity⁴.

Knapper, som hører til vores gitter, skal ses som hørende til gitteret, og knapperne i menuen skal ses som hørende sammen. Figur 2.5, side 8 viser et eksempel på, hvordan vi holder knapperne tæt på gitteret, så de lader til at høre til gitteret.

Vi anvender en menubar som et fast element i den øverste del af brugergrænsefladen, så vi gør det nemt for brugeren at navigere rundt i systemet. Man er aldrig mere end to eller tre klik væk fra listen over lokaler eller ens bookninger. Figur 2.5 viser desuden, hvordan vi har placeret vores menubar.

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

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 fordrag 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 større problemer:

Punkt 1: Forvirring omkring valgte tidspunkter.

Punkt 2: Forvirring angående navigering imellem skærbilleder.

Punkt 3: Forvirring omkring sletning af forplejning/udstyr og booking.

Punkt 4: 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 bruges 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 Vi vurderede, at problemet kunne løses ved hjælp af en FAQ eller hjælpe funktion. Vi mener, at grundstrukturen i vores system er solid og derfor ikke bør ændres kun på grund af dette 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.