**Projektkursus Systemudvikling Trejde delrapport**

04. april 2013

**Instruktor:** Kristoffer Robert Alex Cobley

**Projektgruppe-id:** 8

**Gruppemedlemmer:**

Christian Nielsen – 050190

Allan Nielsen – 070592

Thomas Hansen – 060491

**Review af Ehn og Kyng – Cardboard Computers**Artiklen diskuterer, hvilken rolle Mock-Ups har og hvilken effekt de giver. Herunder redegører de for forskellige måder at lave Mock-Ups på og hvad fordelene/ulemperne er.  
De beskriver 2 måder at lave Mock-Ups på, hhv. *papir- og papir/computer Mock-Up.*   
De gør meget ud af at man benytter papir udgaven, da den er billig mens den stadig giver et realistisk af indblik i, hvordan systemet skal fungere. En såkaldt ”*Hands-on Experience”*.   
Dette skyldes at papir/Computer udgaven udmærker sig ved, at computeren kun skal give en idé om, hvordan systemet ville virke og ikke være systemet. Det har dog den ulempe, at når der er en computer med, kan aktører blive i tvivl og forveksle computerens egentlige egenskab i Mock-Uppet, med ”*The Real Thing*”. Man skal som udgangspunkt benytte en computer i begrænset omfang, således at der ikke sker en forveksling mellem denne ”leg” og det egentlige produkt.  
  
I vores projekt har vi selv brugt Mock-Ups. I forbindelse med vores første møde med kunden, havde vi lavet en Mock-up som et still billede, der skulle give kunden en ide om, hvad vores tanker var i forhold til udseendet af deres system. Vores erfaring er, at det var en rigtig god måde, at kommunikere hvordan vi havde tænkt os flowet i systemet skulle foregå, og ville ikke have undværet den hvis vi skulle gøre det igen.

**Review af No Silver Bullet**Artiklen diskuterer hvordan software industriens problemer med dårligt udviklet software, ikke kan løses med et vidundervåben som sølv mod en varulv.

Forfatteren opstiller to begreber, essens og uheld, softwares essens i bedste Aristotele stil, og uheld som i dag er en del af softwareudvikling, men ikke er naturlige dele af den.

Softwares essens er blandt andet kompleksitet. I software er alle (eller ihvertfald stærkt mange) af dets dele forskellige, der er en masse forhold blandt softwarens element der skal tages højde for, i modsætning til vægene i et hus, som blot kan udvides ved at lægge flere mursten på.

Der opstilles desuden forventninger til sotware, at det kan forme sig efter dets omgivelser, mens dette ikke altid er muligt for det bedste resultat. Derudover ændres kravene til software ofte, under udvikling såvel som efter. Dette er sandt for det meste successfuldt software, folk afprøver det i scenarier det ikke var designet til, og ønsker derfor ændringer til dets feature vifte. Brooks hævder at softwares usynlighed, medfører at det er umuligt at skabe passende modeller af softwaren, det ender i en blanding af en masse forskellige modeller der alle viser noget forskelligt.

Hvilket bringer os til de førnævnte uheld. Brooks omtaler en række opfindelser, som ikke er en naturlig del af software udvikling, men som har haft et stort indtryk på diciplinen. Der er høj-niveau sprog, som medførte øget produktivitet, idet det nu var blevet nemmere for udviklere at skrive software i et sprog der ligner vores talte sprog mere, i forhold til maskin instruktioner som kunne være svære at holde styr på. Dette er dog ikke en pletfri mirakel kur, som Brooks skriver, kan sprogene til tider komme i vejen. Time-sharing gjorde tiden mellem kode skrevet færdigt, og at se det kørt væsentlig mindre. Dette medførte en dekopling mellem tankerne der gik ind i koden, og en trend hvor at når der blev trykket på compile knappen, tømtes hovedet i et splitsekund, som kunne medføre tab af overblik. Samlede programmeringsmiljøer har affødt et økesystem hvor programmer kan arbejde sammen, og senere udvikling af deciderede værktøjssamlinger til software udvikling.

Brooks omtaler herefter en række ting, som blandt andre anses for at være software verdenens “silver bullet”. Herunder, kunstig intelligens, og hvordan det skulle medføre ekspert systemet, som gør det muligt for softwaren at tage selvstændige beslutninger, baseret på et allerede defineret regelsæt. Derudover, automatiseret programmering, hvor at specificere et problem skulle sætte en computer i stand til at udvikle et program som kan løse dette problem. Brooks går videre og snakker om programmering gennem grafisk flader, program verifikation samt miljøer, herunder arbejdsstationer som hardware, og værktøjer som kan tilbyde alt fra syntax highlighting til kode dokumentation.

Den første del af artiklen maler et ret dystert billede af software verdenen, at vi er havnet i en afgrund som vi ikke kan komme ud af. Jeg er af den overbevisning, at software udvikling endnu er på et tidligt stadie. Ikke alle forstår hvad software reelt er, og hvordan det kommer til verden. Det er derfor vores opgave som software udviklere, at bygge bro sådan at vi kan hjælpe brugere med at få udviklet software som passer til deres behov. Som Brooks afslutningsvis lægger vægt på, har vi brug for at sætte mere fokus på design. Det indledende stadie hvor softwaren formes efter de mange krav, sådan at alt spiller som det skal. Det kan her være oplagt, som Brooks også nævner, at se markedet igennem for allerede eksisterende software, måtte det være færdige programmer, til enkelte moduler, som løser vores problem. Der er ingen grund til at opfinde den dybe tallerken til hvert projekt, som adskillige offentlige danske projekter også har vist[[1]](#footnote-1). For at hjælpe brugeren med at afklare behov, kan tests med prototyper med fordel benyttes, som i IBMs ADS projekt[[2]](#footnote-2). Samt slutteligt at ændre vores syn på hvordan software skabes. At software er en process hvor et produkt kultiveres, så det gror, i stedet for at blive bygget af klodser (klasser og metoder).

I vores eget projekt har vi gjort meget brug af den kultiverende process af et produkt. Systemets funktioner bygges på, efterhånden som basale funktionaliteter er færdige. Dette medfører en meget naturlig process, hvor man så at sige for systemet til at kravle, før det går.

1. **Abstract**

Vores gruppe har taget kontakt til skønhedssalonen Nailbeauty. De efterspørger en ekstra feature på deres hjemmeside, i form af et online bookingsystem. Sammen med dette, skal vi også lave et interface, hvori de kan håndtere medarbejdernes kalender, kunder og tider der er booket. Til at implementere dette, benytter vi os af PHP og MySQL.

1. **IT-projektets formål og rammer**

Systemets hovedfunktion er at gøre aktiviteten at booke en tid hos salonen, fuldstændig  
automatisk. Dvs. En kunde skal kunne vælge hvilken behandling der ønskes, hos hvilken medarbejder, hvorefter systemet vil præsentere kunden for ledige tidsintervaller.

Medarbejderne kan se bookede og ubookede tider gennem en traditionel kalender, hvor der ligeledes er mulighed for check-in, redigering og sletning af bookninger. Der tilbydes et interface til medarbejder administration, herunder arbejdstider og behandlinger denne kan foretage, samt et interface til at administrere behandlinger.

Medarbejdernes arbejdsgange skal ændres så lidt som muligt, og der tages hensyn til medarbejdernes såvel som kundernes begrænsede tekniske erfaring.

Systemet skal køres på salonens nuværende webserver, og udvikles i PHP og MySQL. Hvis muligt skal behandlings administrationen integreres i deres nuværende Joomla CMS, sådan at en behandling ikke skal ændres to seperate steder.

Systemets brugere består delvis af salonens kunder, og deres medarbejdere, og disses forskellige tilgang til bookning/bookede tider.

Systemets primære ansvarsområde er at håndtere størstedelen af alle bookninger uden en medarbejders indvirkning.

**3. Kravspecifikation for IT-løsningen** (a)

Lige nu benytter NailBeauty sig af et analogt system. De har en logbog, hvori de manuelt skal skrive hvilken uge de er i, dato på dagene, medarbejdere og ikke mindst kundernes aftaler. Det har visse fordele, de kan f.eks. redigere i aftaler mm. hurtigt og let, sætte flueben så de ved kunden er ankommet. Dette har også ulemper i form af, at de skal være tilstede for at en kunde kan booke en tid, hvilket kun er inden for salonens åbningstider. Bookingsystemet vil gøre at kunder kan booke tider, lige når de har lyst, uanset hvornår på døgnet det er. I deres interface vil de gerne beholde den brugervenlighed og de features der er, for at simplificere overgangen mellem det analoge og digitale.

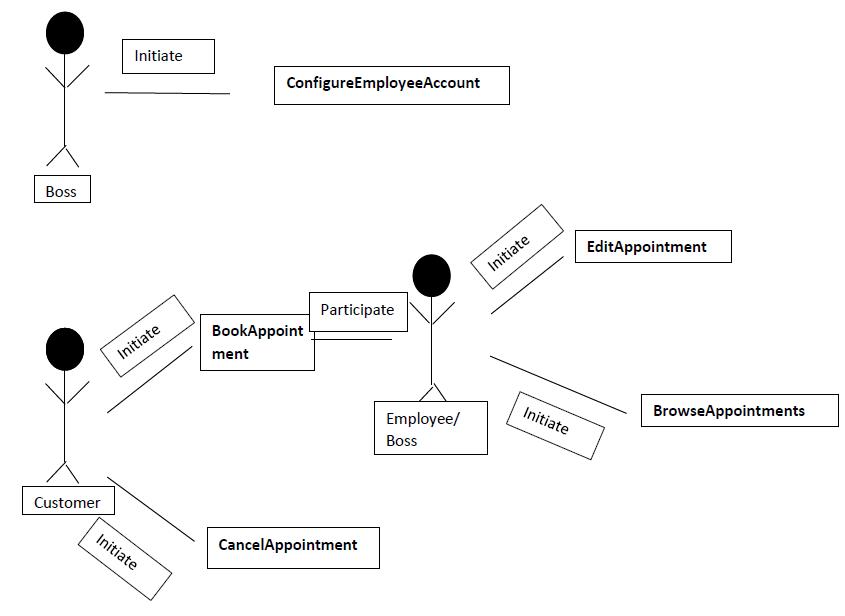
Funktionelle krav:

* Kunde – skal udfra valg af medarbejder, behandling og dato blive præsenteret med en række mulige tider til evt. booking.
* Chefen – skal have samme retigheder som medarbejderen, samt kunne tilføje og fjerne medarbejdere.
* Medarbejder – skal kunne se en oversigt over bookede tider, samt kunne redigere i bookede tider.
* Database – skal indeholde al information om de bookede tider, samt kunde/medarbejder information.

Ikke-funktionelle krav:

* Brugbarhed – Brugergrænsefladen skal være let overskuelig og i overensstemmelse med NailBeauty’s nuværende hjemmeside design. Et enkelt design sikrer en let og overskuelig brugbarhed for selv en ny bruger, derfor ønskes valg af medarbejder, behandling og dato vist på samme side.
* Vedligeholdelse – God dokumentation og konsekvent kode-stil vil sikre at udefrakommende let vil kunne opdatere, vedligeholde samt udvikle systemet.

1. Højniveau use case model:



**ConfigureEmployeeAccount:**

Chefen kan tilføje/fjerne medarbejder information, til ansættelse/fyring.

**BookAppointment:**

Kunden kan booke en tid til en behandling.

**CancelAppointment:**

Kunden kan afbestille en allerede bookede tid.

**EditAppointment:**

Både medarbejdere og chefen kan tilføje og fjerne bookede tider.

**BrowseAppointments:**

Både medarbejdere og chefen kan browse alle bookede tider.

1. Tre specificerede use cases:

**Use case navn:** BookAppointment

**Medvirkende aktører:** Klient

**Flow of events:**

* 1. Klient vælger behandling, en eller flere medarbejdere og kalenderen updateres (benytter use case UpdateCalender).
  2. Klient vælger en ledig dag og tid for behandlingen (benytter use case PickDate).
  3. Hvis flere medarbejdere er valgt og ledige, bedes klienten om at specificere, hvilken medarbjeder vedkommende vil behandles af.
  4. Klienten bekræfter salon kriterier.

**Entry condition:**

* + Klient logger ind I systemet.

**Exit conditions:**

* + Klient har lavet en aftale ELLER
  + Klienten arborterer

\*\* DETALJERET USE CASE\*\*

**Use case navn:** UpdateCalendar

**Medvirkende aktører:** Kalender

**Flow of events:**

1. Systemet opdaterer kalendere, for at vise steder den valgte behandling passer in I forhold til tiden  
The system updates the calendars to show spots where

the selected treatment fits in regards to time, based on.

**Entry condition:**

* Klient er logget ind i systemet.
* Klienten har valgt en behandling og en eller flere medarnejdere.

**Exit conditions:**

* Kalenderen er blevet opdateret.

**Use case navn:** PickDate

**Medvirkende aktører:** Klient

**Flow of events:**

1. Kalenderen er blevet opdateret, for at vise ledige datoer, baseret på forgående valg.
2. Klient vælger en ledig dato

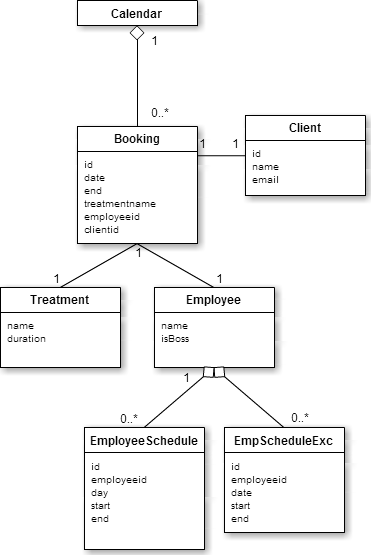
**Entry condition:**

* Klient er logget ind i systemet.
* Klient har valgt en behandling og en eller flere medarbejdere.

**Exit conditions:**

* Klient har valgt en dato.

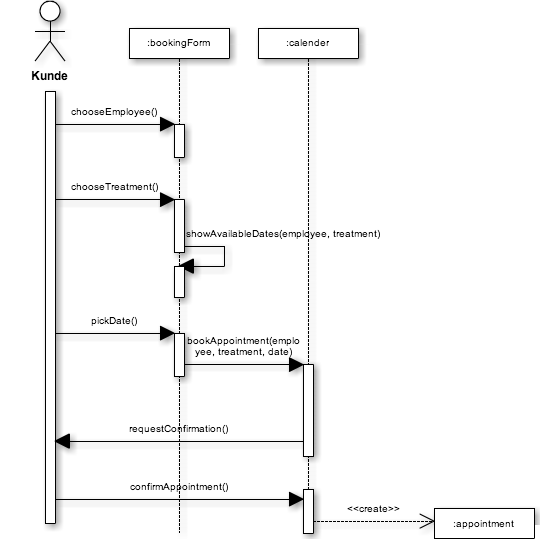
1. **Klassediagram**



Systemet består af en kalender, som er opbygget af en række bookninger. Til hver bookning er der tilknyttet en medarbejder, kunde, behandling og dato. Det er muligt for medarbejderne at se og redigere alle bookninger, og for kunder at se deres personlige bookninger. En medarbejder markeres chef, som gør det muligt at administrere tilbudte behandlinger og medarbejdere. Til hver medarbejder er knyttet en række arbejdsdage, og kan have tilknyttet en række undtagelser til disse, fx kan en medarbejder have en lægeaftale onsdag d. 15-05-2013 mellem 12 og 13, hvorfor der ikke skal være mulighed for at booke en tid hos denne medarbejder i den periode.

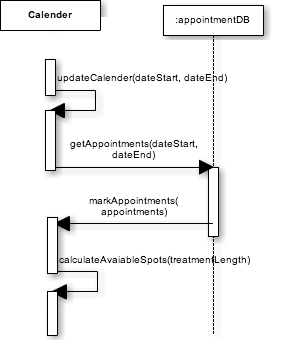
1. **Sekvens-diagrammer**

Use case 1 – BookAppointment



Kunden vælger medarbejder og behandling, derved opdaterer systemet den viste kalender, for at illustrere ledige tider. Kunden kan derefter vælge en dato, hvorefter systemet tjekker om den valgte kombination er gyldig. Hvis ja, så bed om en bekræftigelse og opret bookning.

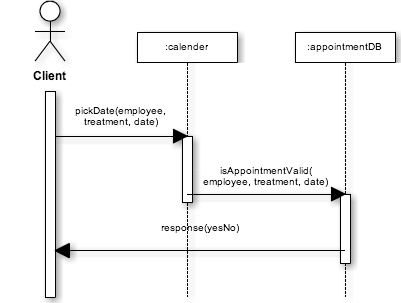
Use case 2 – UpdateCalender



Kalenderen henter alle bookninger indenfor et interval og viser dem i kundens kalender.

Dernæst opdateres kalenderen til kun at vise gyldige tider i.fht. behandlingens varighed.

Use case 3 – PickDate



Kunden vælger en dato, hvorefter systemet tjekker om det er en gyldig combination, og sender feedback til kunden.

**4. Systemdesign Sammenfatning**Vores systemdesign består af en database, hvor data omkring kunder, medarbejdere og aftaler kan hentes af kalenderne i de forskellige instanser (medarbejder, kunde el. chef).   
Det vigtigste er at kunne få kalenderen til at fungere og hente og opdatere data korrekt til/fra databasen, således at f.eks. aftaler ikke kan overlappe hos en medarbjeder.

Systemet har på nuværende tidspunkt implementeret kalenderen, med funktionalitet til booking af tider, i en sådan grad at den ikke tillader overlapninger. Det er muligt for chefen at administrere behandlinger og medarbejdere. Kunde funktionalitet i dets helhed mangler, dvs. mulighed for at specificere hvem bookningen foretages af. Ligeledes mangler der bedre navigeringsmuligheder i kalenderen (navigering i tid), samt mulighed for at medarbejdere kan tjekke ankomne kunder ind, og til slut mulighed for at medarbejderen kan oprette ”exceptions” i sin kalender (fx hvis medarbejderen ikke er tilgængelig onsdag mellem 10 og 12, skal det ikke være muligt at oprette bookinger der.)

**5. Program og systemtest**Test af systemet har hidtil været primitive funktionalitets tests. Når en funktion er blevet tilføjet, tjekkes den ved hånden i browseren, og der afgøres om en funktion virker efter hensigten, eller ej. Planen er på nuværende tidspunkt er at fortsætte i denne bane.

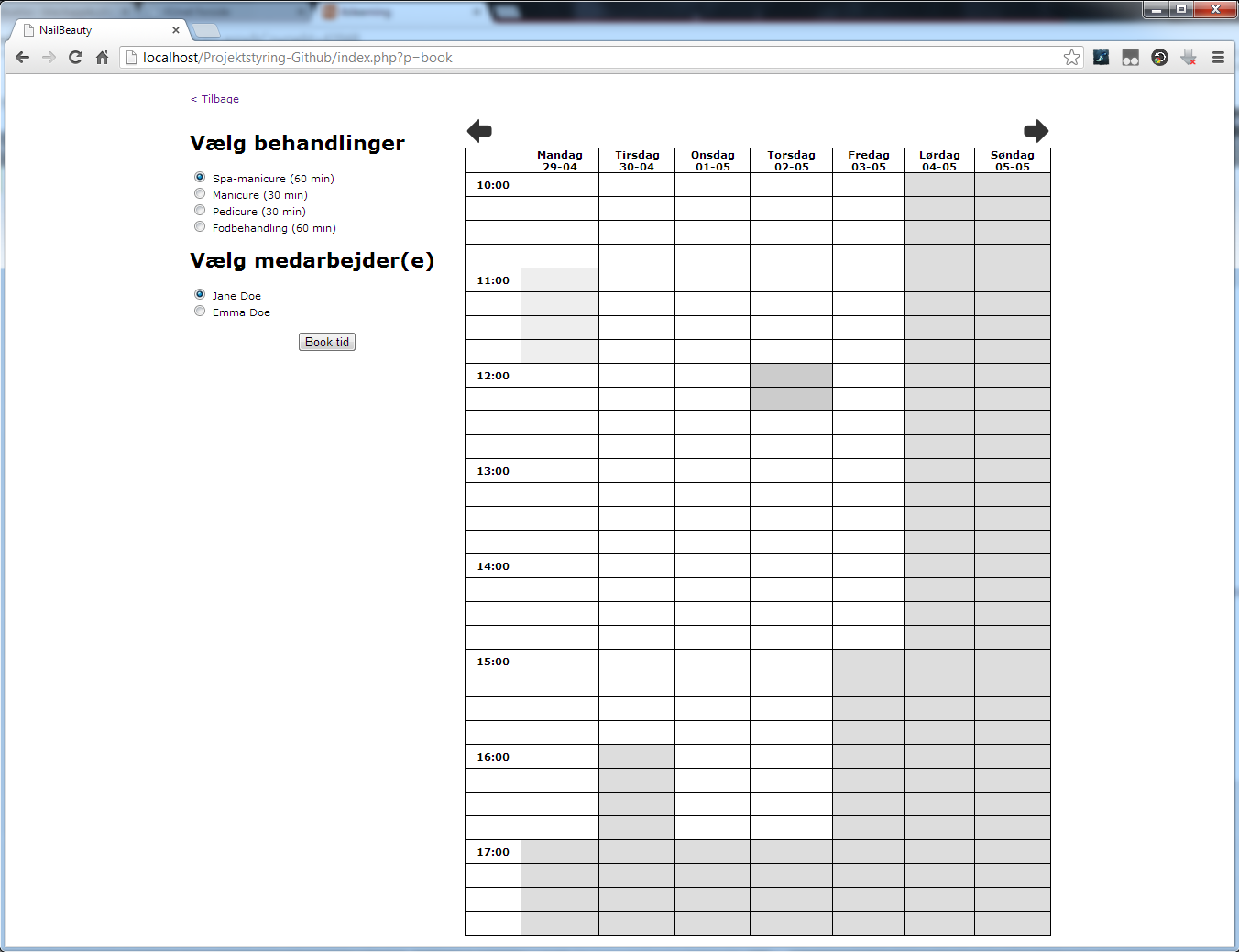
Der er planlagt user-tests ved at bede personer i salondens målgruppe (kvinder), om at udføre en bookning efter særlige specifikationer. Given behandling og muligvis ugiven medarbejder, og et specificeret tidsrum, sådan at det er op til testpersonen at finde en tidsramme der er passende. Testpersonerne er på nuværende stadie, personer fra salonens kundekreds (dette er endnu ikke bekræftet som en mulighed), samt undertøttelse fra vores egne omgangskredse som en backup plan.

Det optimale i dette tilfælde ville være tests med brugere fra salonens nuværende kundekreds, men given sparsom kontakt fra salonens side og deadlines der nærmer sig, ser det ud til vi må nøjes med den mere sub-optimale løsning at bruge vores egen omgangskreds som brugere. Dette betyder at vi højst sandsynligt tester på nogle brugere som er mere vante til computer brug end ellers.

**6. Brugergrænseflade og interaktiondesign**

På nuværende stadie består brugergrænsefladen af en kalender med paneler til valg af behandling og medarbejder. Det er gennem kalenderen muligt at vælge en tidsramme. Kalenderen er inddelt i kvarterer, og kalender-celler markeres når musen holdes over, så brugeren får feedback om hvilken tidsramme der holdes over. Antallet af markerede celler justeres automatisk, sådan at en behandling på 60 minutter fylder 4 celler, og 30 minutter fylder 2 celler. Klikkes der vælges tidsrammen, og brugeren kan fortsætte til en bekræftelse af bookningen.

Det er ikke muligt at vælge de grå celler. Medarbejderen er enten optaget eller ikke på arbejde på dette tidspunkt.



Der kan her ses at brugeren har valgt en 30 minutters behandling torsdag kl 12.00, mens der overvejes en 60 minutters behandling kl 11.00 om mandagen. Trykkes der book tid, bookes den 30 minutters behandling. Skærmbilleder til dette er endnu ikke lavet.

Resten af systemet er demo versioner af medarbejder back-enden, hvor det er muligt at administrere medarbjedere.



**7. Versionstyring**Der er siden sidste rapport blevet udviklet en reel demo. Kalender modulet er i et stadie hvor det er brugbart, og de resterende dele er på et stadie hvor deres funktionalitet kan fremvises.

commit 3a54508d66dc349e1c87c1e92882278d154af1cc

Author: Christian Nielsen <blackapple.chr@gmail.com>

Date: Mon Apr 22 16:04:17 2013 +0200

Updatet employeeform and employeemanager

Improved form element for time picking in employeeform

Added insert, update and delete functionality to employeemanager

commit 047e3db7a186b14b6d4a39c1f6e840164ddb7f15

Author: Christian Nielsen <blackapple.chr@gmail.com>

Date: Sun Apr 21 17:36:16 2013 +0200

Updated calendar and employeeform

Updated calendar to handle schedule in seconds

Updated employeeform to allow insertions (lacks functionality to handle

2 different schedule rangesin same day)

commit c118b334a60aa53ad664ba9cb613973da6383af5

Author: Christian Nielsen <blackapple.chr@gmail.com>

Date: Sat Apr 20 20:35:04 2013 +0200

Updated page loading and added employee managing

Made index.php load page files into itself through url variables

Added employee managing, can serve a list of employees and a link to

edit said employee

Added forms for adding/updating employees in database - lacks code to

interact with db

commit 868feaeadddbb97b4a969dfe8e4dfe197050aed2

Author: Christian Nielsen <blackapple.chr@gmail.com>

Date: Fri Apr 19 18:25:35 2013 +0200

Updated calendar

Added navigation for calendar to browse forwards or backwards 1 week

Added saving to database possible again

commit 9fb32e6a0f3ced9db9935cb6f0d0950890c30acf

Author: Christian Nielsen <blackapple.chr@gmail.com>

Date: Thu Apr 18 20:47:30 2013 +0200

Updated calendar

Added treatment and employee support, calendar now handles employees

work schedules and already booked time slots

commit fa7717b1bf3b226102e1d426dc8d90874235f7ba

Author: Christian Nielsen <blackapple.chr@gmail.com>

Date: Thu Apr 18 00:05:13 2013 +0200

Updated calendar

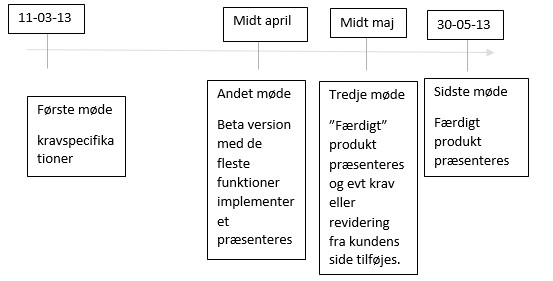
Improved time tracking for bookings

Added mouseover and click effects

Added Database class

**8. Projektsamarbejdet**

Samarbejdet i gruppen er udemærket, vi mødes et par gange om ugen for at snakke og diskutere projektet med hinanden. Derudover har vi haft nogle udviklingssessioner, hvor vi har siddet sammen og udviklet på projektet. Herudover har vi holdt nogle uddybende møder for at holde alle i gruppen ajour med projektet og diskutere fremtidige udfordringer og opgaver. Efter hvert møde skrives et fælles mødereferat, da denne arbejdsform fungerer bedst for os. I referatet gennemgås dagsorden for mødet, beslutninger vi har taget, opgaver der skal laves samt agendaen til næste møde gennemgås.

Gruppen har holdt et møde med kunden, dels for at specificere kundens krav og ønsker dels for at indblik i arbejdsgangen for bedre at kunne opfylde deres behov som virksomhed. Indtil videre har gruppen holdt ét møde med kunden og der er planlagt yderligere tre senere i forløbet. 

Grundet gruppens størrelse har vi besluttet ikke at have en desideret projektleder, men istedet at komme med inputs alle tre. Vi aftaler fælles møder, der afsluttes med et fælles mødereferat. Gennem ugentlig kontakt sikrer vi at alle i gruppen er underforstået med de nuværende opgaver samt har et overblik over projektet i nærmeste skaber det en fælles ide om det færdige produkt. Særligt fungerer de fælles kode-sessioner, hvilket sikrer at alle har indblik og forståelse for produktet samt evt spørgsmål eller uenigheder kan diskuteres og termineres på stedet. Det kan til tder være vanskeligt at holde fokus i adskillige timer ad gangen og dette er især svært når vi skal kode. Derfor kunne det optimeres, at have kodesessioner oftere, men af mindre varighed

**Bilag 1 - Mødereferat:**

**Dato/Tid:** 2013-03-27 12-14

**Deltagere:** Thomas, Allan, Christian

**Dagsorden:**

* Aftale hvilke funktioner der skal indgå i første udkast af det kørende program.
* Gennemgå mødet med kunden, Nail Beauty, så alle er indforstået med kundens krav.
* Gennemgå rapport kravene til del aflevering 2.

**Beslutninger:**

* I det første udkast af det kørende program lægges vægt på den mest fundamentale side af programmet, bookingsiden hvor behandling, medarbejder og kalender præsenteres.
* Valg i det kørende program skal kunne registreres i en database.

**Opgaver:**

* Start programmering af booking systemet, samt opsætning af database.
* Uddelligér basale opgaver til rapporten.

**Agenda til Næste Møde:**

* Aftale hvad der skal implementeres i systemet.
* Aftale nyt møde med Nail Beauty.

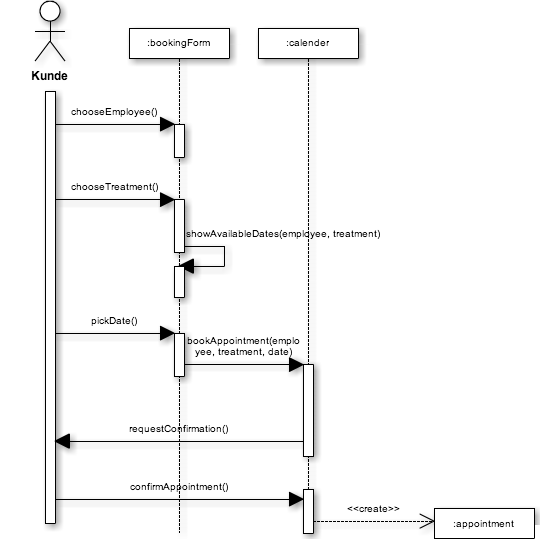
**Næste Møde:** 2013-04-16 kl. 12.00-14.00

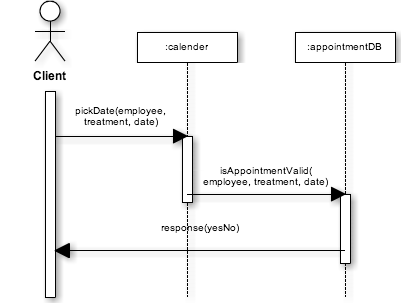
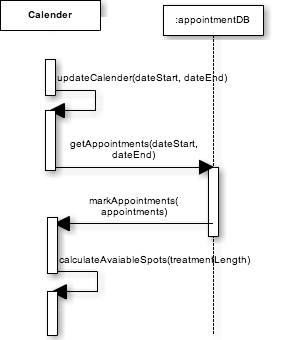
**Bilag 2 Nail Beauty møde**

Præsentation af Bookingsystem for Nail Beauty

* Praktiske oplysninger o Er der nogen der vedligeholder jeres webside? Hvem?
  + Skandia Web – 4452 1900 o Ifb. Bookingsystemet, hvordan vil I have jeres blik ind I kalenderen, hvordan skal det se ud?
* Kolonne pr. Medarbejder
* Check in
* Ingen hård sletning -> egen oversigt
* Note felt fra medarbejder og kunde  Flere behandlinger til samme kunde?
* ja
  + Mor-datter booking?
* Kunde skal ringe
  + Er jeres arbejdstider faste, eller kan de rykkes? Dvs. kan bookinger overskride arbejdstiderne?
* Nej, kræver opringning
* Hvordan ser jeres nuværende bookingsystem ud? Hvordan bliver tider booket?
* SMS varselssystem vil man skulle betale ca. 30 øre/SMS.
* Logning af valgte behandlinger/medarbejdere
* Adgangsbestemte behandlinger
* Uge numre I kalendre
* Mulighed for medarbejdre at oprette tider for kunder
* Spærring af tider til stam kunder
* Startside med betingelser
* Kalendre med kvarters nøjagtighed

**Bilag 4 Use Case 1 (BookAppointment)**



**Bilag 5 Use Case 2 (UpdateCalender) Bilag 6 Use Case 3 (PickDate)** 

1. Rejsekortet, Digital Tinglysning, DSBs IC4, Polsag, o.lign. [↑](#footnote-ref-1)
2. Designing for Usability, John D. Gould & Clayton Lewis [↑](#footnote-ref-2)