

---

# **BRGA**

---

## Central Monitoring System User Interface (CMONUI)

*Frank Henningsen                      20034454*

*Jens Peter Troelsen                    20034441*

*Claus Kirkegaard Clausen 20034435*

**"If they can't use it, they won't use it!"**

<http://www.theusabilitycompany.com/resources/usability.html>

**Group De Luxe**

## 1 Resume

Denne opgave er anden opgave i faget BRuger Grænseflader til Apparater(BRGA). I første iteration fokuserede vi på at udvikle en brugergrænse flade til brug på en touchscreen, men i denne anden iteration har vi fokuseret på at udvikle en grænseflade med talegenkendelse og kunstig tale. Til begge iterationer har vi brugt Cognitive Walkthrough og Heuristic Evaluation til at evaluere brugergrænsefladen. Desuden har vi i første iteration foretaget en keystroke analysis, og i anden har vi udført en brugertest på Århus Kommunehospital, afd. D2.

Formålet med opgaven er at udvikle et Central MONitor System User Interface(CMONUI). CMONUI er en brugergrænseflade til projektet "Central Monitoring System" (CMON), som bliver lavet i faget DRTS<sup>1</sup>. CMON's opgave er at indsamle og præsentere data fra en række klienter. Klienterne blev udviklet i faget IRTS<sup>2</sup>, og de er Lokale patient monitorer(LMON). LMON var et projekt, som gik ud på at måle en patients tilstand på et sygehus. Det vil sige ECG, EDR og puls, samt lave en alarm, hvis et af disse signaler blev kritisk. LMON kører på en SBC686.

Udviklingen og design af brugergrænsefladen laves ved hjælp af teknikker lært i undervisningen, hvor vi især ligger vægt på keystroke analysis, cognitive walkthrough, heuristic evaluation og usability evaluation (brugertest).

CMONUI er optimeret til en touchscreen og besidder mulighed for styring via talegenkendelse. Programmet er udviklet til at køre på windowsplatformen og er implementeret ved hjælp af C#.

---

<sup>1</sup> DRTS: Faget Distribuerede Realtids Systemer, ved Finn Overgaard Hansen - <http://kurser.iha.dk/eit/tidrts/>

<sup>2</sup> IRTS: Faget Indlejrede Realtids Systemer, ved Finn Overgaard Hansen - <http://kurser.iha.dk/eit/tiirts/>

## 2 Indholdsfortegnelse

<b>1</b>	<b>RESUME.....</b>	<b>2</b>
<b>2</b>	<b>INDHOLDSFORTEGNELSE.....</b>	<b>3</b>
<b>3</b>	<b>PROJEKTBEKRIVELSE .....</b>	<b>4</b>
3.1	LOCAL PATIENT MONITERING SYSTEM (LMON).....	4
3.2	CENTRAL MONITERING SYSTEM (CMON).....	5
3.3	CENTRAL MONITERING SYSTEM USER INTERFACE (CMONUI).....	5
<b>4</b>	<b>DESIGN OVERVEJELSER.....</b>	<b>6</b>
<b>5</b>	<b>BESKRIVELSE AF FORLØBET.....</b>	<b>7</b>
5.1	FORUNDERSØGELSE.....	7
5.2	KRAV TIL SYSTEMETS BRUGBARHED .....	7
5.3	VALG AF EVALUERINGSTEKNIKKER.....	8
5.4	PROTOTYPE(R) .....	8
5.5	EVALUERING AF BRUGERGRÆNSEFLADE.....	9
5.5.1	<i>Cognitive Walkthrough</i> .....	9
5.5.2	<i>Heuristic Evaluation</i> .....	11
5.5.3	<i>Keystroke Analysis</i> .....	12
5.5.4	<i>Brugertest</i> .....	14
<b>6</b>	<b>SYSTEMETS BRUGERGRÆNSEFLADER.....</b>	<b>16</b>
6.1	TOUCHSCREEN BRUGERGRÆNSEFLADEN.....	16
6.2	TALE BRUGERGRÆNSEFLADEN .....	17
6.2.1	<i>Tale tilbagemeldinger</i> .....	18
6.2.2	<i>Tale Kommandoer</i> .....	18
6.3	MOBILITY.....	19
<b>7</b>	<b>KONKLUSION.....</b>	<b>21</b>
<b>8</b>	<b>REFERENCER.....</b>	<b>23</b>

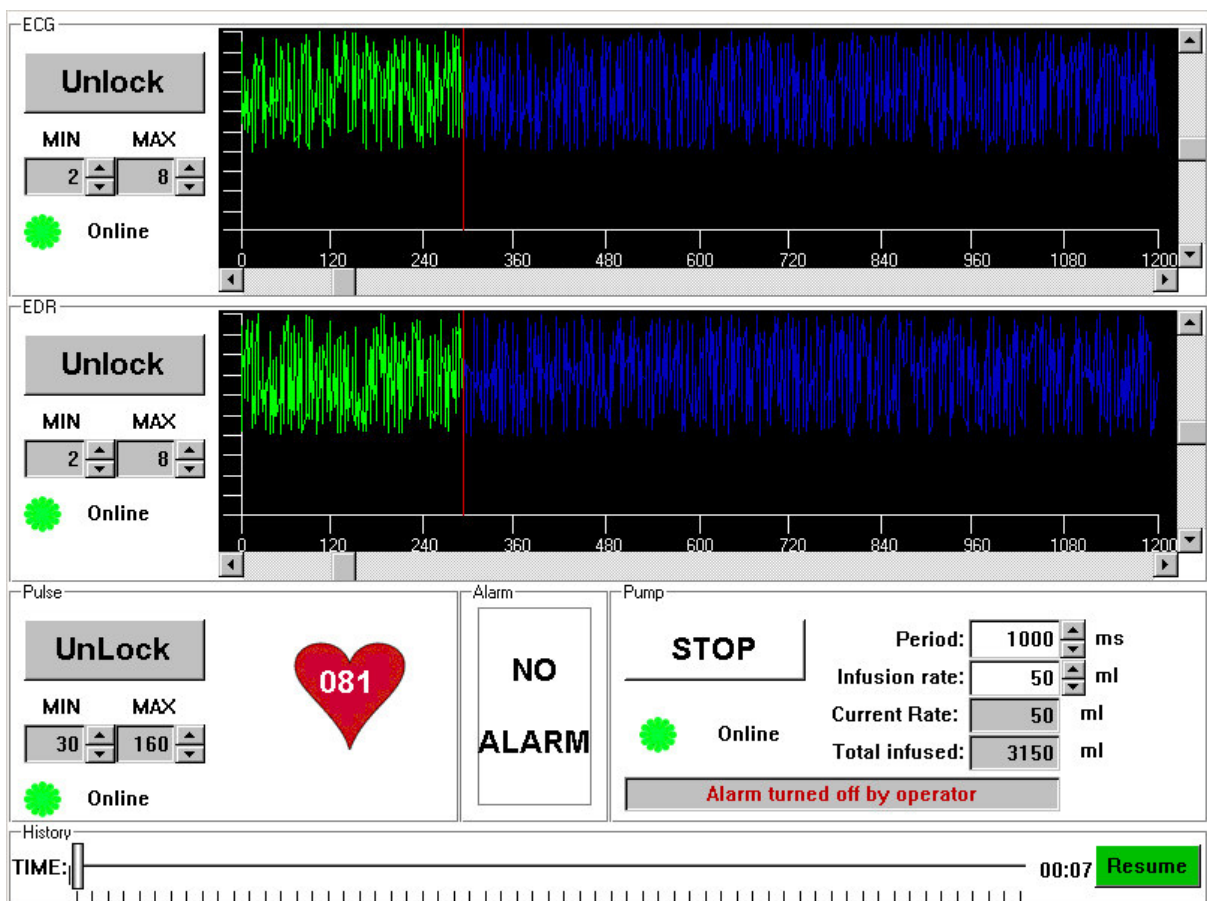
### 3 Projektbeskrivelse

Central MONitor System User Interface (CMONUI), er en del af et større projekt. Ud over det vi laver i faget BRuger Grænseflader til Apparater (BRGA), består projektet af et Local Patient Monitoring system (LMON), og et Central Monitoring System (CMON). Figur 2 viser en samlet oversigt over hele systemet.

#### 3.1 Local Patient Monitoring system (LMON)

LMON systemet blev udviklet i IRTS kurset sidste semester. LMON er et patientmonitorsystem, beregnet til brug i hospitalssektoren, som modtager signaler fra en patient og en pumpe. Fra patienten modtages ECG, EDR og puls signaler, der efterfølgende vises på LMON's display. LMON skal holde øje med patientens grænseværdier, og give alarm hvis disse overskrides. Et skærbillede af systemet er vist i figur 1.

Opgavens problemstilling var, at udvikle et indlejret realtidssystem hvor der tages højde for de problemstillinger der er, ved udvikling til realtidssystemer.



- Figur 1 viser et skærbillede over LMON systemet

### 3.2 Central Monitoring System (CMON)

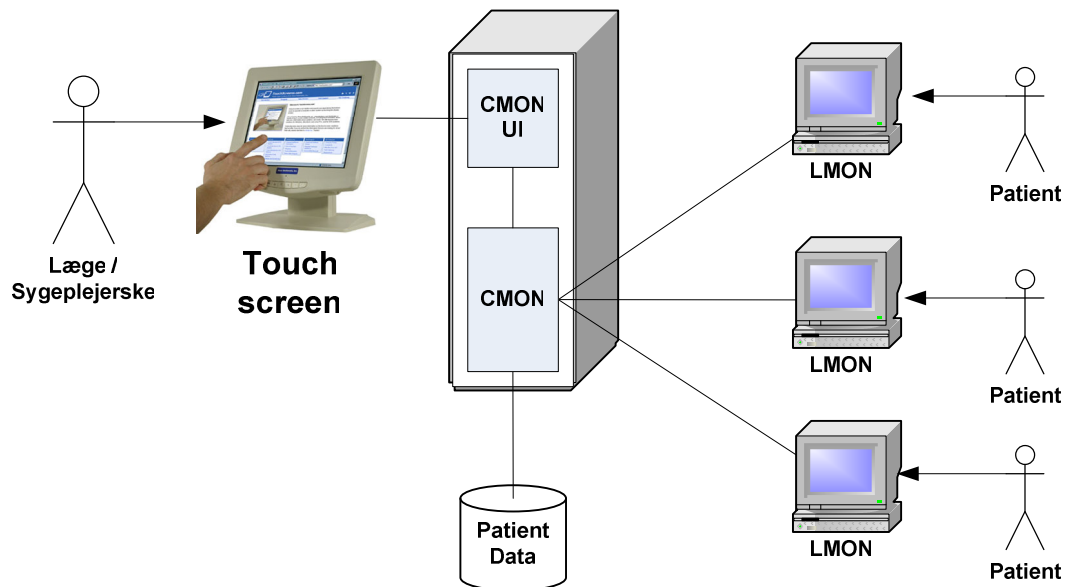
CMON bliver udviklet sideløbende med projektet CMONUI, og er en udvidelse til LMON projektet. CMON er en central server, som overvåger en række LMON klienter. I CMON ligges der vægt på begreber som distribuerede realtids systemer, og design mønstre til distribuerede systemer.

CMON´s opgave er at overvåge en række patienter tilkoblet til LMON systemer, og stille patienternes data til rådighed for CMONUI.

### 3.3 Central Monitoring System User Interface (CMONUI)

CMONUI er udviklet i forbindelse med BRGA faget, og er en brugergrænseflade til CMON. På CMONUI skal det være muligt for en sygeplejerske at se en liste over alle patienter der overvåges af LMON, og ud fra den liste skal det være muligt at vælge hvilke patienter der skal overvåges centralt. På overvågningssiden vises hver enkelt signals tilstand, rød lampe for alarm, og grøn hvis der ikke er nogen alarm. Hvis en bruger ønsker at se samtlige af en patients data klikkes der på brugeren, og man ryger til patientdatasiden. Her skal alle data der vises på den valgte patients LMON display (figur 1) også være tilgængelige på serveren. Vi vil lave brugergrænsefladen så brugervenlig som muligt, ved blandt andet at benytte os af teknikker som keystroke analysis, Cognitive walkthrough, Heuristic evaluation og en test med de virkelige brugere af systemet.

Da CMONUI indgår i et større system, viser vi i figur 2 nedenfor en oversigt over hele systemet.



Figur 2 viser en samlet oversigt over systemets arkitektur

## 4 Design overvejelser

CMONUI er udviklet til at kunne køre på en windows-maskine. Vi havde ligeledes noget underliggende kommunikation der skulle kunne integreres med brugergrænsefladen.

Vi havde valget mellem at implementere CMONUI i MFC eller .NET. Vi valgte sidstnævnte, da vi havde mest erfaring med udvikling på .NET platformen.

Kommunikationen imellem CMONUI og CMON foregår ved at kapsle CMON serveren ind i en COM komponent. Kommunikationen imellem CMON og CMONUI, vil dog først blive lavet efter obligatorisk opgave 1 i BRGA er afleveret. For ikke at skulle vente på at CMON projektet, har vi lavet en passende grænseflade mellem CMONUI og CMON, der gør det muligt at kompensere for en manglende CMON server, ved at lave en stub af CMON. Vi har valgt at bruge Model View Controller (MVC) pattern [POSA1-01], for at få en god opdeling af ansvar og for at adskille data fra præsentationen.

CMONUI udvikles med henblik på at skulle afvikles på en touchscreen. Det vil sige, knapperne, teksten etc. skal være af en størrelse, så brugerne kan benytte fingrene til navigationen. Vi har taget udgangspunkt i retningslinjerne givet i [ELO-04], hvor der er listet 10 guidelines til udvikling af brugergrænseflade til touchscreen apparater.

En ting vi fandt meget vigtig, var at det skal være hurtigt og nemt at betjene. Brugeren skal på en simpel måde kunne overvåge nogle udvalgte patienter og nemt danne sig et overblik over disse patienters tilstand.

En anden design overvejelse vi har gjort os er, at programmets farvevalg skal harmonere med det miljø der er på et hospital.

Vi har taget højde for at CMONUI skal kunne udvides med talegenkendelse, i kraft af vores valg af MVC pattern. For at udvide systemet til også at omfatte talegenkendelse, behøver vi kun at tilføje en Voice User Interface (VUI) controller samt at give controlleren tilgang til vores model data.

## 5 Beskrivelse af forløbet

Under udvikling af CMONUI har vi forsøgt at gøre brug af nogle af de teknikker vi har lært i BRGA faget. Vi har primært taget udgangspunkt i [Nielsen-93] teknikker, men også undersøgt andre metoder som B. Schneiderman's [Schneiderman-98]. På trods af at der findes mange forskellige teknikker, lavet af forskellige personer, er mange af principperne de samme. Der er f.eks. bred enighed om at en brugergrænseflade bør være konsistent.

Vi har benyttet [Nielsen-93 s. 72] udviklingsmodel som udgangspunkt, til at sammensætte vores udviklingsforløb. Vores udviklings forløb har været delt op i flg. faser:

- For undersøgelse
- Specificere krav til systemets brugbarhed
- Valg af evalueringsteknikker
- Udvikle prototype(r)
- Evaluering af brugergrænseflade ud fra udvalgte evalueringsteknikker
- Iterativ design af brugergrænseflade

Vi har gjort brug af en iterativ udviklingsmodel, hvor vi løbende har evalueret og revideret brugergrænsefladen. Vi vil i nu kort beskrive hver fase i udviklingsforløbet.

### 5.1 Forundersøgelse

Vi startede projektet med at analysere og undersøge de forskellige evalueringsteknikker. Dette er foregået ved at følge undervisningen i BRGA kurset, samt brug af Internet for finde andre artikler omkring emnet. Vi var ikke på forhånd bekendt med den store mængde teori omkring Human Computer Interface (HCI), og det var derfor nødvendigt at undersøge de forskellige muligheder og teknikker før vi kunne specificere vores krav til systemets brugbarhed.

### 5.2 Krav til systemets brugbarhed

Efter vi havde valgt et projekt til BRGA kurset (CMONUI), har vi specificeret krav til systemets brugbarhed. Systemets brugbarhed i denne kontekst omhandler systemets brugergrænseflade og brugervenlighed.

Vores krav til systemet kommer i første omgang fra opgave-formuleringen, hvor der er stillet krav om udvikling af et touchscreen baseret system og et system med brug af kunstig tale og talegenkendelse.

Vores eget krav til systemet er at lave et "Walk-Up-And-Use" system, som vil kunne anvendes uden større oplæring. Vi vil under hele forløbet fokusere på at samtlige brugergrænseflader er så brugervenlige, så brugeren ikke kan lave fejl. Dette vil vi gøre ved kun at tilbyde et minimum af

funktioner i en bestemt rækkefølge, og på denne måde undgå at brugeren kommer til at lave en forkert handling.

Det vil dog blive nødvendigt med oplæring i forbindelse med talegenkende, da de mulige tale kommandoer ikke er vist på brugergrænsefladen.

### 5.3 Valg af evalueringsteknikker

Heuristikker bruges til at lave et godt design af brugergrænseflader. Der findes mange forskellige heuristikker [Nielsen-93][Schneidermann-98][ELO-02], og for hver brugergrænseflade er det op til designeren at vælge de heuristikker der bedst passer til brugergrænsefladen.

I første iteration valgte vi at gøre brug af 3 teknikker til evaluering af systemets brugbarhed: Cognitive Walkthrough (CW), Heuristic Evaluation (HE) og Key-stroke-analysis (KSA).

I anden iteration valgte vi ligeledes en lave CW, samt to HE'er: en af touchscreen grænsefladen og en anden af talegenkendelse grænsefladen.

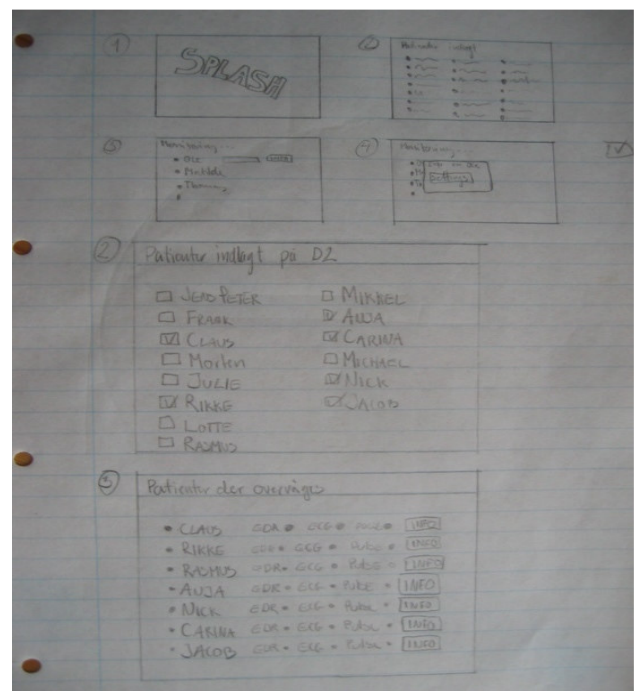
Desuden har vi valgt at lave en brugertest [Nielsen-93] med systemets slutbrugere, for at afklare hvordan systemet bliver brugt og om der er eventuelle uoverensstemmelser.

Vi vil starte med at foretage en CW af vores prototype, for så tidligt som muligt at indikere eventuelle problemer i brugergrænsefladen. Når prototypen er blevet revideret starter vi en iterativ udviklingsproces, hvor vi løbende vil afholde en HE af systemet. Sidst i forløbet vil vi anvende en Keystroke Analysis (KSA) der bygger på GOMS og Keystroke-Level Model (KSM) som beskrevet i [MacKenzie-03] [Sneiderman-98 s. 55], for at undersøge om systemets ydeevne er acceptabelt.

Efter vi har lavet HE, CW og KSA, vil vi udføre en brugertest for at finde de fejl, som typiske brugere af systemet vil møde.

### 5.4 Prototype(r)

Vi har tidligt i forløbet lavet en skitse over hvordan systemet skal opbygges. Vi havde forestillet os at brugergrænsefladen bestod af 4 dele: Splash, Overview, Detail og Patient-View. Vores intention var at lave brugergrænsefladen så brugeren ikke kunne lave en fejl, ved kun at tilbyde et minimum af muligheder. Dette valg er naturligvis på bekostning af fleksibilitet, som beskrevet af [Odlyzko-99 s. 55].



• Figur 3 viser første prototype



Figur 3 viser vores første udkast til en prototype lavet på en tavle og nedskrevet på papir. Denne skitse viser i store træk hvordan brugergrænsefladen kommer til at se ud.

Efterfølgende har vi implementeret en prototype i C#, der er blevet anvendt under evaluering i 1 iteration. Denne ses på figur 4.

Under udviklingen af denne prototype, blev layout og design løbende blevet diskuteret og rettet til.

I forbindelse med anden iteration og de første usability tests, blev brugergrænsefladen endnu en gang lavet om. De centrale elementer blev bibeholdt, men farver, symboler og tekst blev ændret. Resultatet af disse ændringer, kan ses i figur 5.

Name	ECG	EDR	Pulse	Pump
Claus Kirkegaard Clausen	Green	Green	Green	Green
Jens Peter Troelsen	Green	Green	Green	Green
Frank Henningsen	Green	Green	Green	Green
John Doe	Green	Red	Green	Green
Foo Bar	Green	Green	Green	Green
S. YgMand	Green	Green	Green	Green
Jens Larsen	Green	Green	Green	Green
Pim der kaiser van der blabla	Green	Green	Green	Green
S. YgMand	Green	Green	Green	Green
EndnuS. YgMand	Green	Green	Green	Green
And AS. YgMand	Green	Green	Green	Green

• Figur 4 viser C# Prototypen i 1 iteration

Valgt	Navn	CPR
✓	Jens Peter Troelsen	123456-7890
✓	Claus Clausen	123456-7890
✓	Frank Henningsen	311274-1234
✓	Henning Pimpdahl	123456-7890
✓	Never Ending Story	123456-7890
✓	Gerda Gern	123456-7809
✗	Jørgen Nisseballe	123456-8890
✗	Marie Nielsen	123456-7897
✗	Annemarie Johnsen	123456-7770
✗	Caroline Caspersen	123456-7889
✗	Kasper Jørgensen	123456-7887

• Figur 5 viser C# Prototypen i 2 iteration

## 5.5 Evaluering af brugergrænseflade

I de følgende afsnit vil vi beskrive de teknikker vi har brugt til at evaluere vores brugergrænseflader. Hver metode vil kort blive beskrevet efterfulgt af en diskussion over observationerne fundet under evaluering.

### 5.5.1 Cognitive Walkthrough

En CW er en teknik til at evaluere et systems brugbarhed ved at analysere brugernes opfattelse af systemet [Lewis-97 s 333]. CW's opgave er at afklare om et system er let at bruge/lære [Abowd-95] [CW-Drexel], og dokumentere eventuelle problemer under evaluering. Nogle af fordelene ved CW er at der ikke behøves at være slut-brugere til afvikling, og at man tidligt i udviklingsforløbet kan udføre evalueringen. En ulempe ved CW er at testen kun finder 40% af de fejl, man ellers ville have fundet under en slut-bruger test [Lewis-97][Riihiaho-00].

En CW består af 4 trin [BRGA-6 s. 337]:

1. Specificere systemets brugere og deres baggrund
2. Specificere systemets opgave(r)
3. Specificere hver opgaves korrekte sæt af handlinger
4. Specificere brugergrænsefladens tilstand under afvikling af en opgaves handlinger

For hver handling i en opgave skal nedenstående spørgsmål dokumenteres:

- Vil brugeren forsøge at opnå handlingens ønskede effekt ? (Havde brugeren samme intention som udvikleren)
- Er brugeren i stand til at foretage den korrekte handling ?
- Vil brugeren være sikker på at den handling han forsøger at opnå er korrekt ?
- Vil brugeren forstå systemets tilbage melding ?

Desuden skal eventuelle fejl og mangler dokumenteres i fejl rapporten.

For at kunne lave en CW, har vi lavet en skabelon bestående af 3 dele: Forberedelse, Evaluering og Fejl rapport (bilag 1). Forberedelsen har vi udfyldt i fællesskab, og indeholder en liste over systemets opgaver og de handlinger der aktiverer opgavens korrekte effekt. For hver CW er det testpersonens opgave at udfylde en evaluerings- og fejl-rapport, og dokumentere resultater fundet i testen.

I første iteration havde vi specificeret 3 opgaver i vores CW formular, men i forbindelse med overgangen til anden iteration er der kommet en ekstra opgave med: "Ændre øvre grænseværdi for ECG signalet"

I vores CW formular har vi specificeret 4 opgaver og deres korrekte sæt af handlinger:

<b>Mål / Opgaver</b>	
	<ol style="list-style-type: none"><li>1. Se Alle Patienter<ol style="list-style-type: none"><li>a. Tryk på 'Se Alle Patienter' knap</li></ol></li><li>2. Overvåg Udvalgte Patienter<ol style="list-style-type: none"><li>a. Vælg Patient(er)</li><li>b. Tryk på 'Udvalgte Patienter' knap</li></ol></li><li>3. Vis Patient Data<ol style="list-style-type: none"><li>a. Vælg Patient(er)</li><li>b. Tryk på 'Udvalgte Patienter' knap</li><li>c. Tryk på den ønskede patient</li></ol></li><li>4. Ændre øvre grænseværdi for ECG signalet<ol style="list-style-type: none"><li>a. Vælg Patient(er)</li><li>b. Tryk på 'Udvalgte Patienter' knap</li><li>c. Tryk på den ønskede patient</li><li>d. Tryk på den øverste 'Pil-op' knap, ved ECG grafen</li></ol></li></ol>

Grunden til at der er så få handlinger i opgaverne, skyldtes vores "Walk-Up-And-Use" design krav, der bevirker at brugeren har et minimum af muligheder under afvikling.

### 5.5.1.1 CW observationer

Vi har foretaget en CW i både 1 og 2 iteration, og vi har begge gange kunnet konstatere problemer i vores brugergrænseflade. Resultaterne fundet i første iteration drejede sig primært om at tilføje systemet mere hjælpe information og tilbagemeldinger. I anden iteration fandt vi væsentligt færre fejl end tidligere, og de problemer vi fandt omhandlede hovedsageligt PatientView panelet, der på trods af store design anstrengelser præsenterede for mange informationer, til at man hurtigt kunne få et overblik.

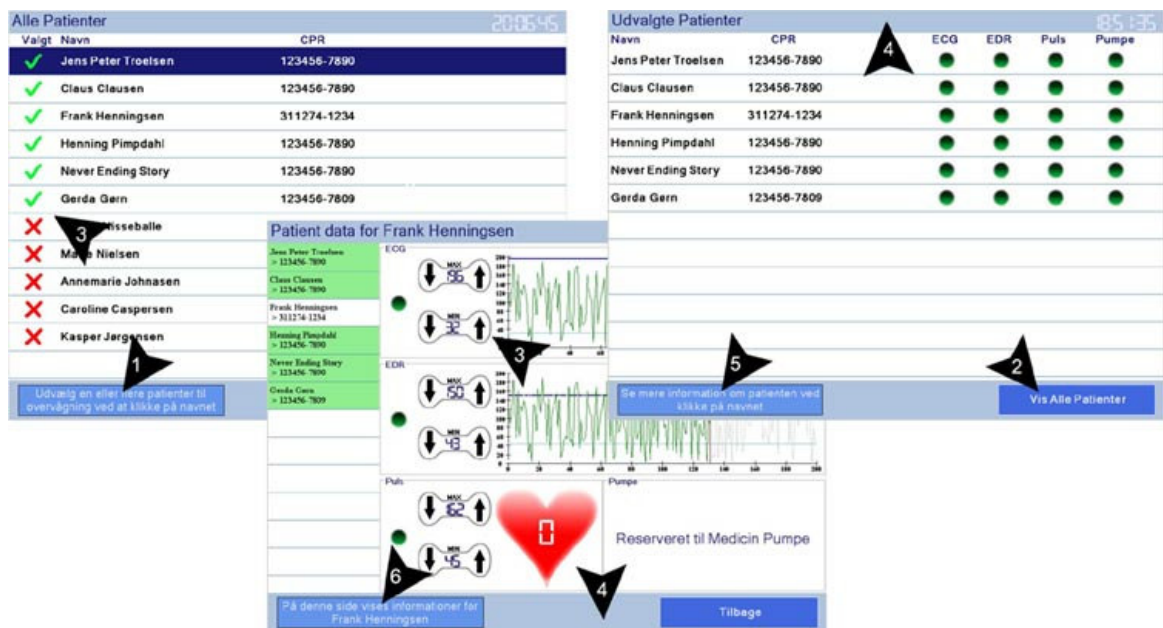
### 5.5.2 Heuristic Evaluation

En HE er en teknik til at evaluere et systems brugbarhed ved at analysere om brugergrænsefladen overholder gængse retningslinjer for god brugergrænseflade design. En HE er en systematisk gennemgang af brugergrænsefladen ud fra en række relevante heuristikker, som f.eks. Niensens 10 [Nielsen-93 s. 20] eller Shneiderman's 8 [Sneiderman-98 s. 74] gyldne regler.

For at kunne lave en HE har vi udvalgt de heuristikker vi mener er relevante for vores system. Vi har taget udgangspunkt i retningslinjerne fra [Nielsen-93 s. 20], der definerer 10 retningslinjer for hvordan et godt brugergrænseflade design skal se ud. Ud af disse 10 retningslinjer har vi fundet 6 der har betydning for vores brugergrænseflade, disse er beskrevet i nedenstående tabel.

Nr	Heuristik	Beskrivelse
1	Brug simple og naturlige dialogbokse	Undgå urelevant information
2	Tal brugerens 'sprog'	Sørg for at bruger forstår systemets tilbagemeldinger, og at teksten er klar og entydig
3	Minimer brugerens betænkningstid	Sørg for tydeligt at markere muligheder, og gøre brug af let genkendelige symboler
4	Vær konsistent	Undgå tvetydighed
5	Brug informative tilbagemeldinger	Systemets status skal vises, og respons tid skal være lav
6	Hjælp og dokumentation	Tilbyd hjælp

For at illustrere hvordan vi opfylder ovenstående heuristikker, viser vi i figur 6 nedenfor prototypens 3 brugergrænseflader med pile ud for de guidelines de overholder.



- Figur 6 viser systemets 3 brugergrænseflader med pile ud for de guidelines de overholder

For at kunne dokumentere resultaterne fundet ved HE, har vi lavet en skabelon bestående af 2 dele: Forberedelse og Test (bilag 2). Første del har vi lavet i fællesskab og indeholder liste over de anvendte heuristikker, mens anden del skal udfyldes af testpersonen .

### 5.5.2.1 HE observationer

Ligesom vi gjorde med CW har vi foretaget en HE i både 1 og 2 iteration. I anden iteration har vi udført en HE af både touchscreen- og voice- brugergrænsefladen, da de resultater man finder under en HE afhænger af den brugergrænseflade man undersøger.

Under HE i første iteration gjorde vi store observationer, og fandt frem til en meget vigtig problemstilling, vi ikke tidligere havde tænkt på. Vi fandt at prototypen i første iteration skjulte alle patienters tilstand på nær den patient man overvågede via PatientView panelet. Dette var ikke acceptabelt, og vi måtte finde en løsning på at vise en patients vitale tilstand samtidig med at man kunne se alle andre patienters tilstand samtidig.

I anden iteration fandt vi stadig mange fejl i vores brugergrænseflade, men denne gang havde problemerne skiftet karakter, og drejede sig primært om at tilføje systemet flere sikkerhedsmekanismer. Vi fandt at der var mange sikkerhedshensyn at tage i forbindelse apparater på et sygehus, og at disse skulle undersøges. En af mulighederne vi diskuterede var brug af iris- eller finger- scanner, som mulig autentifikations mekanisme.

### 5.5.3 Keystroke Analysis

Det er vigtigt at man hurtigt kan navigere rundt i prototypen, da det er et overvågnings system til patienter på et sygehus. For at undersøge hvor hurtigt vi kan navigere rundt, har vi i forbindelse med

første iteration valgt at bruge Keystroke-Level Model(KLM) metoden[MackKenzie-03 s 3], der kan måle hvor hurtigt et givent scenario er. En KLM beregning er summen af den tid det tager at udføre et givent scenario, hvor scenariet bliver opdelt i tastetryk (K), pointing (P), homing (H), drawing (D), mental operator (M) og systemets responstid (R).

Til KLM analysen af prototypen fra første iteration havde vi valgt 3 scenarier:

Scenarie	<ol style="list-style-type: none"> <li>1. Se Alle Patienter             <ol style="list-style-type: none"> <li>a. Tryk på 'Se Alle Patienter' knap</li> </ol> </li> <li>2. Overvåg Udvalgte Patienter             <ol style="list-style-type: none"> <li>a. Vælg 5 Patienter</li> <li>b. Tryk på 'Udvalgte Patienter' knap</li> </ol> </li> <li>3. Vis Patient Data             <ol style="list-style-type: none"> <li>a. Vælg Patient(er)</li> <li>b. Tryk på 'Udvalgte Patienter' knap</li> <li>c. Tryk på den ønskede patient</li> </ol> </li> </ol>
----------	---

Til de forskellige events har vi tilføjet gennemsnits tider: Tiderne er vi kommet frem til ved at gå scenarierne igennem 10 gange og tage gennemsnits tiderne af de forskellige events.

Opgaver	Tid (sek.)
Tastetryk 'Se alle patienter', 'Udvalgte patienter' og 'Hjælp'	0,1
Vælg patient(er)	0,9
Tryk på den ønskede patient	0,5
M for at vælge en patient	1,0
M for at få vist patient data	0,9
Responstid fra systemet, til samtlige events	0,08

Da prototypen har en touchscreen brugergrænseflade, er tastetryk, og pointing slået sammen i  $t_K$ .

### 5.5.3.1 KSA observationer

Scenarie	Mellemregning ( $T_{execute} = t_K + t_P + t_H + t_D + t_M + t_R$ )	Tid (sek.)
1. Se Alle Patienter	$T_{execute} = 1 * t_K + 1 * t_R = 0,1 + 0,08$	0,18
2. Overvåg Udvalgte Patienter	$T_{execute} = 5 * t_K + 1 * t_K + 1 * t_M + 6 * t_R = 5 * 0,9 + 0,1 + 1,0 + 6 * 0,08$	10,4
3. Vis Patient Data	$T_{execute} = 1 * t_K + 1 * t_K + 1 * t_M + 2 * t_R = 0,1 + 0,5 + 0,9 + 2 * 0,08$	1,66

Resultatet viser at scenariet 'Overvåg udvalgte patienter', tager relativt lang tid i forhold til de andre scenarier. Dette scenarie udføres i opstarten af overvågningen, hvor alle patienter er vist for brugeren, hvilket giver mange input at tage stilling til. Da scenariet kun udføres sjældent og der ikke er andre/bedre muligheder til at liste alle patienter, er tiden acceptabel. Efter scenariet 'Overvåg udvalgte

patienter' er valgt, vises kun de udvalgte patienter, hvilket giver et bedre overblik, og medfører at scenariet 'Vis Patient Data' kan udføres meget hurtigt.

#### 5.5.4 Brugertest

Tidligt i 2 iteration har vi foretaget en brugertest af brugergrænsen, med prototypen udviklet i forbindelse med første iteration. Brugertesten blev udført som en 'Think Aloud' test [Nielsen-93], hvor brugerne "tænker" højt. Think Aloud metoden har til opgave at fortælle udviklerne hvordan systemets brugerne opfatter systemet, og er ifølge [Nielsen-93 s195] en af de bedste usability metoder.

Brugertesten blev udført i to gange, med forskellige brugere; en sygeplejerskestuderende samt en gruppe af sygeplejersker på Århus Kommunehospital. Dette var to meget forskellige bruger grupper, hvor vi på den ene side havde en sygeplejerskestuderende der på forhånd kendte til brugen af IT, og en gruppe af sygeplejersker på Kommunehospitalet der ikke var vant til brugen af IT, og derfor var mere tilbageholdende.

Testen bestod af to dele; i første del skulle der udvælges 4 patienter der skulle overvåges, anden del bestod i at se en enkelt patients data. Til testen blev der benyttet en tablet-pc der skulle symbolisere en touchscreen-enhed.

Den første forsøgsperson, havde en del kommentarer brugergrænsefladen. En af de største mangler, var CPR-nr. Hun fortalte at der ofte kan være flere patienter med det samme navn, og desuden benytter de CPR-nr'et som reference til patienterne. I PatientView fandt hun at der manglede en tilbage knap for at komme tilbage til oversigten over udvalgte patienter. I prototypen under test var knappen kaldt 'Vis Udvalgte Patienter', og dette var ikke et velvalgt navn.

Der var en smule forvirring om, betydningen af de grønne og røde indikationslamper i oversigten over udvalgte patienter. Dette havde muligvis noget at gøre med, at forsøgspersonen ikke var kendt med begreber som EDR og ECG.

Forsøgsperson havde også en række forslag til forbedringer, herunder øget sikkerhed da

der er tale om personlige data og integration med elektronisk patient journal (EPJ). Den overordnede konklusionen fra brugeren var at systemet var smart.



• Figur 7 viser sygeplejerske studerende Anne i færd med vores bruger test.

Forsøgspersonerne på Århus Kommunehospital havde også en række kommentarer til brugergrænsefladen. For det første syntes de at oversigten over patienter var forvirrende, og at det ikke var klart hvilke patienter der var forbundet til systemet og hvilke der ikke var.

I PatientView kommenterede de ligeledes at der manglede en tilbage knap for at komme til oversigten over udvalgte patienter.

Igen var der tvivl omkring de engelske begreber, EDR og ECG, og igen var der tvivl om hvordan man skulle forholde sig til de røde og grønne indikationslamper i oversigten over udvalgte patienter.

Forsøgspersonernes kommentarer var, at det var et system der var nemt at finde ud af, og at det sikkert kunne benyttes på en intensivafdeling. Desuden informerede de at et tilsvarende system allerede eksisterede på hjerteafdelingen.



- Figur 8 viser gruppen af sygeplejersker der deltog under bruger testen på Århus Kommunehospital.

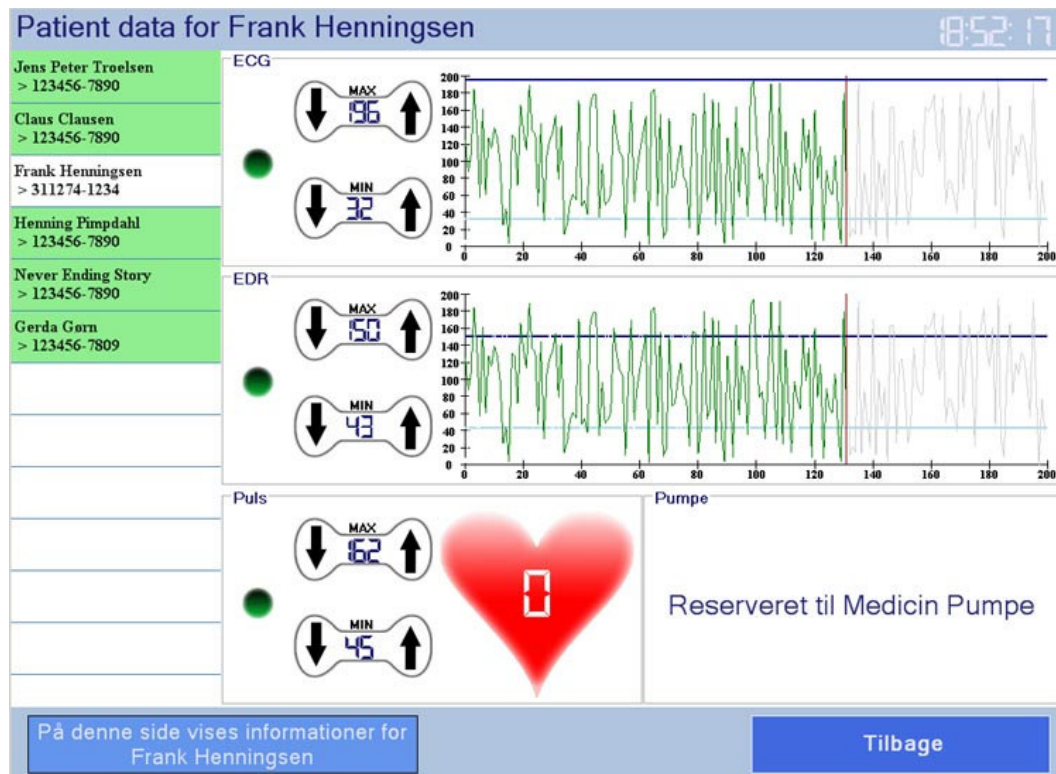


## 6 Systemets Brugergrenseflader

### 6.1 Touchscreen brugergrensefladen

Vi har på grundlag af krav til obligatorisk opgave 1 optimeret vores brugergrenseflade til at blive afviklet på en touchscreen. Denne egenskab har haft store konsekvenser for det overordnede grafiske design. Vi har under hele forløbet forsøgt at overholde retningslinjerne givet i [ELO-04], for at vores touchscreen brugergrenseflade følte god og naturlig. Ud af de retningslinjer opgivet i [ELO-04], har specielt krav 3 om *"Use a simple point-and-click interface with large buttons"* til tider været svær at imødekomme. Vi har under forløbet gentagende gange måttet re-designe vores prototype for at gøre mere plads til større grafiske elementer. På trods af problemer under udvikling er det dog lykkedes hver gang at finde frem til en passende løsning, der tilgodeser kravene til touchscreen brugergrensefladen.

Den mest avancerede og komplicerede del af vores prototype er PatientView panelet vist nedenfor.



- Figur 9 viser et skærbillede over PatientView panelet.

Dette panel har til opgave at vise alle information omkring en patient, og indeholder derfor en del grafiske komponenter. Under udvikling af dette panel har specielt 3 krav været afgørende for det endelige design: [Nielsen-93]’s *"Brug simple og naturlige dialogbokse"* og *"Minimer brugerens betænkningstid"*, samt [ELO-04]’s *"Use a simple point-and-click interface with large buttons"*. Vi har forsøgt at holde os til disse heuristikker samt at lave et simpelt udseende samtidig med at minimere brugerens betænkningstid.



## 6.2 Tale brugergrænsefladen

Vi har til obligatorisk opgave 2 valgt at udvide CMONUI med et Voice User Interface (VUI). I modsætning til CMONUI, som designet til at være et Walk-Up-And-Use system, kræver brugen af VUI en 5 minutters introduktion. Dette skyldes, at det stadig skal være muligt at bruge CMONUI, uden at man ved at det er muligt at snakke til den, da vi skønnede ekstra hjælp information på skærmen ville ødelægge det simple layout. De fem minutters introduktion, skal bruges til at lære brugeren de nødvendige kommandoer, da disse ikke er synlige for brugeren [Sun-98 afsnit 3.3.2].

Formålet med et VUI til CMONUI, er at det stadig skal være muligt for en sygeplejerske/læge, som sidder et stykke væk fra displayet, at bruge CMONUI. Et tænkt scenarie, kan være en sygeplejerske der sidder og laver skrivebords arbejde. I hendes papirer kan hun se en grund til at hr. Hansen på stue 7 skal overvåges. Mens hun sidder ved skrivebordet, og arbejder videre, sætter hun hr. Hansen til overvågning ved hjælp af stemmestyring, og får resultatet tilbage i form af stemmeoutput fra programmet.

Vi har under udvikling af VUI hele tiden haft i mente, at det skulle være et godt brugergrænseflade, og prøvet at overholde de regler og guidelines vi brugte til at designe første version af CMONUI, som CW [Abowd-95] og HE [Nielsen-93 s. 20], for at mindske antallet af fejl, når vi senere skulle gå dem igennem igen til VUI. Vi har også under udviklingen lagt vægt på de design guidelines til stemmestyring der er givet i Java Speech API Programmer's Guide [Sun-98], og ud fra disse retfærdiggjort, at det var en fordel for brugeren, at der blev tilført stemmestyring.

Ud fra Java Speech API Programmer's Guide [Sun-98] ved vi at det er passende at bruge stemmeinput, når:

```
Task requires the user's hands to be occupied so they cannot use a keyboard or mouse (e.g., maintenance and repair, graphics editing).3
```

I vores tilfælde, vil sygeplejerskens/lægens hænder være beskæftiget med andet kontor arbejde.

En anden regel for hvornår det er passende at bruge stemmeoutput lyder:

```
Task requires the user's eyes to be looking at something other than the screen (e.g., driving, maintenance and repair).4
```

Hvor en sygeplejerske/læge i vores tilfælde vil have hendes øjne på skrivebordet, hvor hun er i gang med kontor arbejde.

---

<sup>3</sup> Citat fra [Sun-98 afsnit 3.1 tabel 3.1]

<sup>4</sup> Citat fra [Sun-98 afsnit 3.1 tabel 3.2]

### 6.2.1 Tale tilbagemeldinger

Vores VUI bliver understøttet af et af et grafisk layout, og hver eneste stemmeinput, vil resultere en ændring af layoutet. Således kan en bruger kontrollere at systemet har modtaget tale kommandoen, ved at se om der er en ændring i grafikken. Foruden visuel tilbagemelding vil hvert stemmeinput også resultere i et stemmeoutput, der bekræfter at tale kommandoen er modtaget af systemet. En bruger behøver således ikke at have sine øjne på skærmen hele tiden, men kan styre systemet på afstand via tale kommandoer. I tilfælde af at brugeren kommer til at lave en tale fejl, vil brugeren få oplæst en fejl besked med information om fejllens årsag som beskrevet i [Nielsen-93 s. 20] og [Sneiderman-98 s. 74].

### 6.2.2 Tale Kommandoer

Da vores kommandoer ikke er synlige for brugeren, men kræver 5 minutters oplæring, har vi valgt at lave dem så simple som muligt. Vi har desuden valgt at lave så få kommandoer som muligt, både for ikke at belaste brugeren, og for at forbedre stemmegenkendelsen.

Applications with less *confusable* grammars typically get better accuracy. Similar sounding words are harder to distinguish.<sup>5</sup>

Under et gruppemøde hvor vores prototype kørte i baggrunden, fandt vi dog ud af at kommandoerne var for simple, da vores snak i baggrunden medførte at prototypen reagerede helt vildt, og forstod vores baggrund støj som tale kommandoer. Dette var ikke en acceptabel opførsel. Vi løste problemet ved at lave en kommando, som kunne slå stemmestyring til og fra, på samme måde som lægerne styrer Hermes på Hvidovre hospital[Pol-04].

En liste over systemets tale kommandoer er beskrevet i nedenstående tabel.

Stemme Input	Stemme Output	Resultat
Computer	Yes. Speech recognition enabled. / Yes. Speech recognition disabled.	Slår stemmestyring til/fra. Dette kan gøres alle steder i programmet. Alle andre kommandoer virker kun når stemmestyring er slået til
Select x (Hvor x er et tal)	(Patientens navn) selected.	Virker kun i tilstanden overview Vælger patient nr x.
Select x (Hvor x er et tal)		Virker kun i tilstanden SelectedPatients Skifter tilstand til patientdata, med patient x valgt.
Show selected patients	Selected patients	Skifter view til valgte patienter.
Show all patients	All patients	Skifter view til alle patienter

<sup>5</sup> Citat fra [Sun-98 afsnit 2.2.3]

Stemme kommandoerne er alle på engelsk og kan derfor umiddelbart forstås af computeren uden oplæring. Vi har dog ikke taget højde for at dette kan være et problem for brugerne.

Vi har valgt at lade alt stemmeoutput blive læst op af computerens syntetiske stemme, selvom det lyder unaturligt. Dette skyldes at vi har et dynamisk output, da navnene på indlagte patienter hele tiden skifter, og en blanding af optaget tale, og syntetisk tale vil være for ukonsistent [Sun-98 afsnit 3.3.4].

### 6.3 Mobility

En mulighed vi ikke nåede at undersøge var at lave en mobil brugergrænseflade til systemet. En mobil grænseflade ville kunne tilføje systemet mange nye muligheder, og vil derfor være en naturlig udvidelse i en fremtidig udgave af systemet. Senere vil vi skitsere en mulig løsning for at illustrere nogle af mulighederne.

Da vi på et sygehus kan tale om local mobility [Bardram-Kjær-Nielsen, afsnit 1], hvor brugerne af systemet bevæger sig indenfor et mindre område, vil den mobile enhed således være en del af det større hele, og bruges når en sygeplejerske/læge lettere ville kunne løse en opgave.

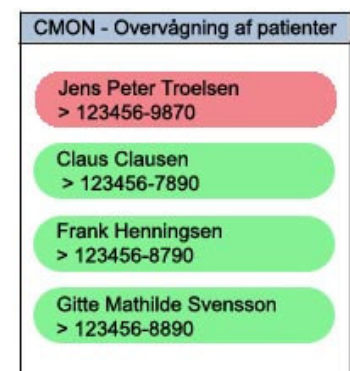
Mængden af information på en afdeling kan være stor. Der kan være mange patienter indlagt, og det kan være uoverskueligt at bladere igennem dem alle på et mobilt apparat. Derfor skal den mobile applikation, selv være i stand til, ud fra hvor den befinder sig, at vise relevante data for brugeren. Som beskrevet af Nielsen og Søndergaard:

```
.... site specific task tasks require site specific information, and thus workers need locally relevant information to get there work done [Nielsen-00, afsnit 1].
```

I vores tilfælde, vil det f.eks. være ideelt for en sygeplejerske at få vist en patients data, når hun går ind på patientens stue og derved få kontekst afhængig information. Denne løsning vil f. eks. kunne implementeres vha. en af de kortrækkende trådløse teknologier som bluetooth eller infrarød kommunikation.

Vi vil nu kort beskrive vores overvejelser omkring en mulig mobil grænseflade til CMONUI systemet.

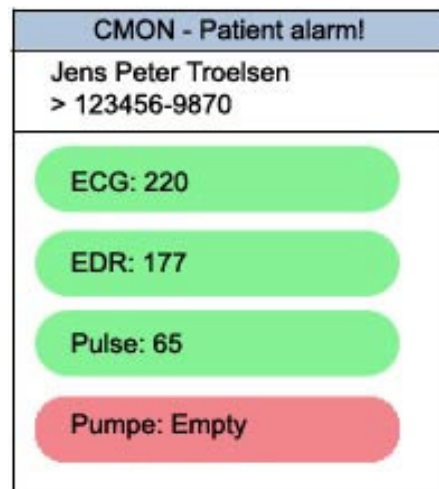
På det mobile interface, skal det ikke være muligt at vælge hvilke patienter der skal overvåges. Her er det tilstrækkeligt at se de patienter der er valgt til overvågning. Det er heller ikke nødvendigt for brugeren at se hvilken type alarm der er, det er nok at vide at der er



- Figur 10 viser en skitse over den mobile grænseflade, hvor en alarm har indtruffet

en alarm. De patienter der er alarm på placeres øverst på listen, og deres navne vil blive røde, samt brugeren vil få en audio tilbagemelding om at der er gået noget galt.

Når brugeren registrerer der er en alarm kan denne bevæge sig til den pågældende stue. Her vil applikationen automatisk registrere at den er i nærheden af en patient og vise patientens vitale data. På det mobile apparat er det ikke muligt at tegne graferne op realtime, både på grund af displayet størrelse, batteri forbrug og CPU kraft. I stedet vises patientens data i form af 3 tal.



- Figur 11 viser en skitse for hvordan den mobile grænseflade viser en patients vitale data

## 7 Konklusion

I dette projekt har vi udviklet en prototype til et patient overvågnings system i et forløb bestående af 2 iterationer. I første iteration fokuserede vi på at lave en touchscreen brugergrænseflade og i anden fokuserede vi på en brugergrænseflade vha. talegenkendelse. Vi har under hele forløbet gjort brug de teknikker vi har lært i forbindelse med BRGA faget, og derved stiftet bekendtskab med teknikker som Cognitive Walkthrough, Heuristic Evaluation, Key-Stroke-Analysis og brugertest i praksis.

Vi mener at have indset både styrker og svagheder i teknikkerne, og det er således vigtigt at gøre brug af de rigtige teknikker på de rigtige tidspunkter.

Vi har ved selvsyn konstateret at Cognitive Walkthrough ikke finder alle fejl, men at det alligevel er en brugbar metode i kraft af at metoden kan udføres tidligt i forløbet og uden indblanding af systemets endelige brugere. De observationer vi gjorde i forbindelse med CW har uden tvivl gjort vores brugergrænseflade mere brugbar og brugervenlig.

Den metode vi brugte mest var Heuristic Evaluation. HE udmærkede sig ved at finde fejl vi ikke tidligere havde fundet, foruden at finde mange af de samme fejl som CW. At HE ligesom CW kan udføres tidligt i forløbet, og uden systemets brugere gør det til en god og billig metode til at forbedre en brugergrænseflade.

En vigtig parameter for hvor mange fejl man finder ved en given evaluering er antal personer med i testen. I vores projekt gruppe var vi kun 3 personer, og vi har alle stort set samme baggrund. Det ville have været fordel hvis vi havde brugt en større og mere varieret bruger gruppe med forskellig baggrund. Foruden en større gruppe ville vi gerne have talt med ekspert brugere, der typisk finder langt flere fejl end alle andre grupper [Nielsen-93][Riihiaho-00].

I vores brugertest blev vi opmærksomme på problemer, vi ikke tidligere havde bemærket. Dette var bl.a. at patientens CPR-nummer altid skal vises sammen med patientens navn, da der kan være flere patienter med det samme navn, på en afdeling.

Blandt kommentarerne under brugertesten var, at det var smart og nemt at bruge. Dette skal dog måske tages med en lille forbehold, da vi på forhånd kendte testpersonerne, og at de derfor ikke ville kritisere det, når vi nu sad på den anden side af bordet, på trods vi opfordrede dem til at være hudløst ærlige.

Vi har udviklet brugergrænsefladen så tekst, knapper og ikoner er så store, at programmet kan afvikles på en touchscreen baseret computer. Et skærmbillede over det endelige system ses nedenfor.

Valgt	Navn	CPR
✓	Jens Peter Troelsen	123456-7890
✓	Claus Clausen	123456-7890
✓	Frank Henningsen	311274-1234
✓	Henning Pimpdahl	123456-7890
✓	Never Ending Story	123456-7890
✓	Gerda Gørn	123456-7809
✗	Jørgen Nisseballe	123456-8890
✗	Marie Nielsen	123456-7897
✗	Annemarie Johnsen	123456-7770
✗	Caroline Caspersen	123456-7889
✗	Kasper Jørgensen	123456-7887

Udvælg en eller flere patienter til overvågning ved at klikke på navnet

Vis Udvalgte Patienter

- Figur 12 viser et skærbillede over den endelige CMONUI brugergrænseflade

Vi mener at det er lykkedes os at udvikle en prototype der indfrier vores krav til brugergrænsefladen. Vores prototype er kendetegnet ved et simpelt design og en stor brugervenlighed og vi er stolte over at have lavet systemet, og glæder os til at gøre brug af prototypen til vores endelige DRTS projekt.

## 8 Referencer

Reference ID	Beskrivelse
[Abowd-95]	Gregory Abowd. 1995. <i>Introduction to Software Engineering</i> . <a href="http://www.cc.gatech.edu/computing/classes/cs3302/documents/cog.walk.html">http://www.cc.gatech.edu/computing/classes/cs3302/documents/cog.walk.html</a> Information omkring Cognitive Walkthrough.
[Bardram-03]	Bardram, J., Kolbeck, T. A. K. and Nielsen, C. (2003). Supporting Local Mobility in Healthcare by Application Roaming among Heterogeneous Devices. In <i>Proceedings of the Fifth International Conference on Human Computer Interaction with Mobile Devices and Services (Mobile HCI'03)</i> . Springer Verlag, 2003. <a href="http://www.daimi.au.dk/~sorsha/Papers/mobile_HCI.pdf">http://www.daimi.au.dk/~sorsha/Papers/mobile_HCI.pdf</a>
[ClosetPlus-97]	ClosetPlus2000: Rhonda Nelson og Melissa Sandlin. 1997. <i>Part 2: Initial Prototype and Evaluation Plan</i> <a href="http://www.cc.gatech.edu/classes/cs6751_97_fall/projects/closet2000+/part2v1.html">http://www.cc.gatech.edu/classes/cs6751_97_fall/projects/closet2000+/part2v1.html</a> Praktisk information omkring Cognitive Walkthrough, samt en række templates.
[CW-Drexel]	Zhijun (William) Zhang. <i>Cognitive Walkthrough</i> . <a href="http://www.pages.drexel.edu/~zwz22/CognWalk.htm">http://www.pages.drexel.edu/~zwz22/CognWalk.htm</a>
[ELO-04]	10 Touchscreen Application Tips. ELOTouch. <a href="http://www.elotouch.com/support/10tips.asp">http://www.elotouch.com/support/10tips.asp</a> 10 guidelines til udvikling af brugergrænseflader til touchscreen apparater.
[FDA.GOV-00]	U.S. Department of Health and Human Services. 2000. <i>Medical Device Use-Safety: Incorporating Human Factors Engineering into Risk Management</i> . <a href="http://www.fda.gov/cdrh/humfac/1497.html#_Toc486653489">http://www.fda.gov/cdrh/humfac/1497.html#_Toc486653489</a> En artikel omkring guidelines til udvikling af apparater til hospitals sektoren. Artiklen brugt af Frank Henningsen til reflektionsafsnit
[Hura-03]	Susan L. Hura. July 2003. <i>Heuristics: Lessons in the Art of Automated Conversation</i> . <a href="http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetspeech/html/hlaac_iv.asp">http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetspeech/html/hlaac_iv.asp</a> Artiklen brugt af Claus Kirkegaard Clausen til reflektionsafsnit
[Lassen-04]	Lee Lassen, Christoffer Magnussen, Jacob Frølund Pedersen 2004. <i>Steril interaktion</i> . <a href="http://www.daimi.au.dk/~ups/PvC/final-projects/09/report/">http://www.daimi.au.dk/~ups/PvC/final-projects/09/report/</a> Artiklen brugt af Jens Peter Troelsen til reflektionsafsnit
[Law-03]	Lai-Chong Law & Ebba Thora Hvannberg 2003. <i>Classification of Usability Problems</i> . <a href="http://citeseer.ist.psu.edu/654197.html">http://citeseer.ist.psu.edu/654197.html</a> Artiklen brugt af Jens Peter Troelsen til reflektionsafsnit
[Lewis-97]	C. Lewis & C. Wharton. 1997. <i>Handbook of Human-Computer Interaction</i> . En bog omkring HCI. God beskrivelse af Cognitive Walkthrough metoden.
[MacKenzie-03]	S. MacKenzie. 2003. <i>Motor Behaviour Models for Human-Computer Interaction. Chapter for J. M. Carroll, Toward a multidisciplinary science of human-computer interaction</i> . <a href="http://www.cs.uta.fi/~scott/mmm/MacKenzieChapter.PDF">http://www.cs.uta.fi/~scott/mmm/MacKenzieChapter.PDF</a>

[Nielsen-93]	Jakob Nielsen. 1993. <i>Usability Engineering</i> . ISBN: 0125184069. Bog om design af brugbare brugergrænseflader.
[Nielsen-00]	Nielsen, C. and Søndergaard, A. (2000). Designing for mobility: an integration approach supporting multiple technologies. In <i>Proceedings of the 1st Nordic Conference on Human-Computer Interaction (NordiCHI 2000)</i> (CD-ROM). <a href="http://www.daimi.au.dk/~sorsha/Papers/NORDICHI_printed.pdf">http://www.daimi.au.dk/~sorsha/Papers/NORDICHI_printed.pdf</a>
[Norman-90]	Norman, D.A 1990. <i>The Design of everyday things</i> Artiklen brugt af Jens Peter Troelsen til reflektionsafsnit
[Odlyzko-99]	A.M. Odlyzko. 1999. <i>The visible problems of the invisible computer: A sceptical look at information appliances</i> . <a href="http://www.firstmonday.org/issues/issue4_9/odlyzko/index.html">http://www.firstmonday.org/issues/issue4_9/odlyzko/index.html</a> Artikel omkring udvikling af brugergrænseflader
[Pedersen-97]	Pedersen, K. & Hansen 1997. <i>Design til skærmen</i> Artiklen brugt af Jens Peter Troelsen til reflektionsafsnit
[Pol-04]	Fra avisen Politiken 1. okt 2004, artikel om Hermes. Online version af artiklen: <a href="http://politiken.dk/VisArtikel.sasp?PageID=338728">http://politiken.dk/VisArtikel.sasp?PageID=338728</a>
[POSA1-01]	Frank Buschmann, Regine Meunier, Hans Rohnert Peter Sommerlad, Michael Stal. 2001. <i>Pattern-Oriented Software Architecture - A System Of Patterns, Volume 1</i> . ISBN: 0471958897 En must-have bog om software patterns.
[Riihiaho-00]	Sirpa Riihiaho. 2000. Experiences with Usability Evaluation. <a href="http://citeseer.ist.psu.edu/riihiaho00experiences.html">http://citeseer.ist.psu.edu/riihiaho00experiences.html</a> En artikel hvor forsk. usability teknikker sammenlignes. Artiklen brugt af Frank Henningsen til reflektionsafsnit
[ Shneiderman-91]	B. Shneiderman. 1991. <i>Touchscreens Now Offer Compelling Uses</i> . <a href="http://citeseer.ist.psu.edu/shneiderman91touchscreens.html">http://citeseer.ist.psu.edu/shneiderman91touchscreens.html</a> Artiklen brugt af Jens Peter Troelsen til reflektionsafsnit
[ Shneiderman-98]	B. Shneiderman. 1998. <i>Designing the User Interface 3<sup>rd</sup> Edition</i> . ISBN 0201694972 Bog om design af brugbare brugergrænseflader. Et alternativ til Jakob Nielsen's metoder.
[Sun-98]	<i>Java™ Speech API Programmer's Guide Version 1.0 - October 26, 1998</i> <a href="http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/index.html">http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/index.html</a>
[Wilson-04]	Chauncey Wilson 2004. <i>Kiosk Design Guidelines</i> . <a href="http://www.stcsig.org/usability/topics/devices.html">http://www.stcsig.org/usability/topics/devices.html</a> Artiklen brugt af Jens Peter Troelsen til reflektionsafsnit
[Yankelovich-95]	Nicole Yankelovich. 1995. <i>Designing SpeechActs: Issues in Speech User Interfaces</i> . <a href="http://citeseer.ist.psu.edu/yankelovich95designing.html">http://citeseer.ist.psu.edu/yankelovich95designing.html</a> Artiklen brugt af Claus Kirkegaard Clausen til reflektionsafsnit
[Webster-99]	Andy Harter, Andy Hopper, Pete Steggles, Any Ward, Paul Webster. 1999. <i>The anatomy if a Context-Aware Application</i> . Cambridge University. <a href="http://www.uk.research.att.com/pub/docs/att/tr.1999.7.pdf">http://www.uk.research.att.com/pub/docs/att/tr.1999.7.pdf</a> En artikel omkring Kontekst-kendte applikationer Artiklen brugt af Frank Henningsen til reflektionsafsnit