



AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI
3. SEMESTERPROJEKT

Dokumentation

Gruppe 4

Mads Fryland Jørgensen (201403827)

Jeppe Tinghøj Honeré (201371186)

Nicoline Hjort Larsen(201405152)

Freja Ramsing Munk (201406736)

Sara-Sofie Staub Kirkeby (201406211)

Tine Skov Nielsen (201404233)

Vejleder

Projektvejleder

Thomas Nielsen

Aarhus Universitet

7. december 2015

Gruppemedlemmer

Mads Fryland Jørgensen (201403827)

Dato

Jeppe Tinghøj Honoré (201371186)

Dato

Freja Ramsing Munk (201406736)

Dato

Nicoline Hjort Larsen (201405152)

Dato

Sara-sofie Staub Kirkeby (201406211)

Dato

Tine Skov Nielsen (201404233)

Dato*Vejleder*

Thomas Nielsen

Dato

Ordliste

Ord	Forklaring
-----	------------

Indholdsfortegnelse

Ordliste	iii
Kapitel 1 Indledning	1
Kapitel 2 Kravspecifikation	3
2.1 Indledning	3
2.2 Funktionelle krav	3
2.2.1 Aktør-kontekst diagram	3
2.2.2 Aktørbeskrivelse	3
2.3 Use cases	5
2.3.1 Use case diagram	5
2.4 Ikke-funktionelle krav	9
2.4.1 (F)URPS+	9
Kapitel 3 Design	13
3.1 Indledning	13
3.2 Hardware arkitektur	13
3.2.1 Design af forstærker	14
3.2.2 Design af analogfilter	15
3.2.3 Grænseflader	18
3.3 Software arkitektur	18
3.3.1 GUI	18
3.3.2 UML klassediagram	21
3.3.3 Applikationsmodel	21
3.4 Software implementering	28
Kapitel 4 Integrationstest	29
4.1 Hardware	29
4.2 Software	29
Kapitel 5 Accepttest	31
5.1 Indledning	31
5.2 Accepttest af Use Cases	31
5.2.1 Use Case 1	31
5.2.2 Use Case 2	32
5.2.3 Use Case 3	33
5.2.4 Use Case 4	33
5.2.5 Use Case 5	33
5.2.6 Use Case 6	34
5.3 Accepttest af ikke-funktionelle krav	35

Bilag	39
Fejlrapport	39
Logbog	39
Mødereferat	39
Kode	39
Samarbejdsaftale	39

Indledning 1

Ansvarsområde

Initialer:

JTH - Jeppe Tinghøj Honoré

TSN - Tine Skov Nielsen

SSK - Sara-Sofie Staub Kirkeby

NHL - Nicoline Hjort Larsen

FRM - Freja Ramsing Munk

MFJ - Mads Fryland Jørgensen

Afsnit Ansvarlig

Kravspecifikation 2

Version	Dato	Ansvarlig	Beskrivelse
---------	------	-----------	-------------

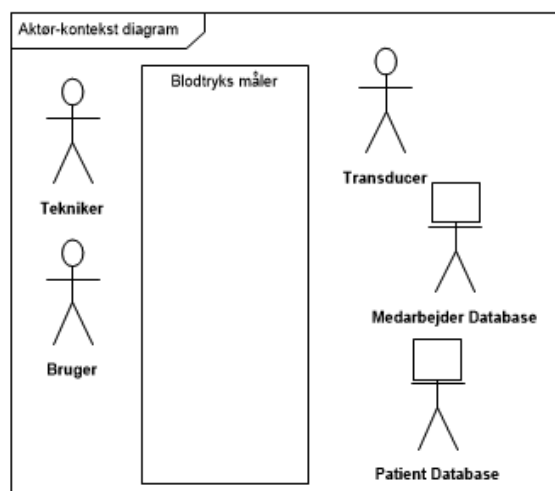
2.1 Indledning

Et blodtryksmålingssystem er blevet udarbejdet, der helt generelt er opbygget af en hardware del og en software del. Hardware delen er opbygget af en forstærker, transducer og analogt filter hvis formål er at kunne forstærke og filtrerer blodtrykssignalet og software delen kalibrerer, nulpunktsjusterer og analyserer signalet samt viser signalet grafisk på en brugergrænseflade.

Kravspecifikationen vil beskrive, produktets kunnen ud fra nogle opstillede funktionelle og ikke funktionelle krav til systemet. De ikke funktionelle krav er krav, der er opstillet til selve systemet og de funktionelle krav er beskrevet i use cases, som omhandler systemts aktiviteter i forhold til aktører og mål.

2.2 Funktionelle krav

2.2.1 Aktør-kontekst diagram



Figur 2.1: Aktør-kontekst diagram

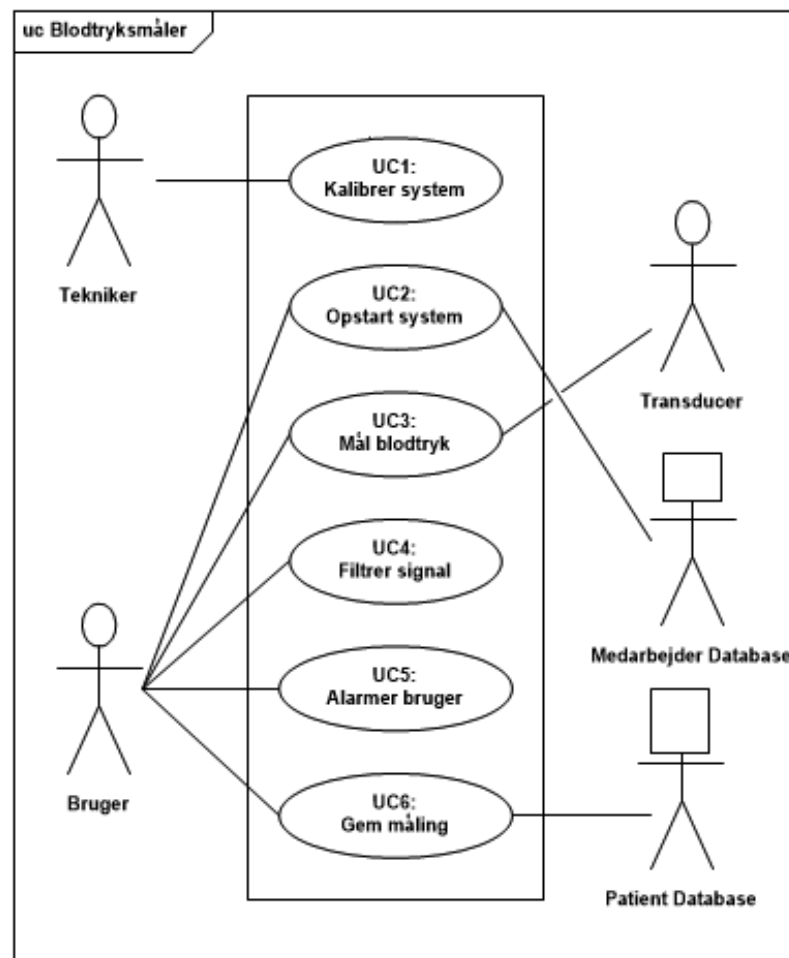
2.2.2 Aktørbeskrivelse

Aktørnavn	Type	Beskrivelse
Bruger	Primær	Brugeren er den aktør der foretager blodtryksmålingerne. Brugeren er en person der har kendskab til systemet, samt tilladelse til at benytte systemet. Fx en læge, anesthesi sygeplejerske
Tekniker	Primær	Tekniker er den aktør der foretager den årlige kalibrering af systemet. Teknikeren er en person der har kendskab til den tekniske del af systemet. Fx. medicotekniker på et sygehus
Transducer	Sekundær	Det er transduceren, der leverer blodtryksmålingen til systemet
Medarbejder database	Sekundær	Medarbejder databasen er det sted, hvor medarbejderens log ind valideres
Patient database	Sekundær	Patient databasen er det sted, hvor blodtryksmålingens data gemmes og patientens CPR nummer valideres

Tabel 2.2: Aktørbeskrivelse

2.3 Use cases

2.3.1 Use case diagram



Figur 2.2: Use case diagram

Use Case 1

Navn	Kalibrer system
Use case ID	1
Samtidige forløb	1
Primær aktør	Tekniker
Initialisere	Tekniker
Mål	Tekniker ønsker at foretage kalibrering
Forudsætninger	
Resultat	Systemet er kalibreret
Hovedforløb	1. Tekniker påtrykker systemet tre kendte tryk

2. Tekniker aflæser responserne
3. Tekniker noterer afvigelser fra de kendte tryk
[3.a Der er ingen afvigelser]
4. Tekniker kalibrerer afvigelsen

Undtagelser	3.a	Use case afsluttes
-------------	-----	--------------------

Tabel 2.3: Fully dressed Use Case 1

Use Case 2

Navn	Opstart system
Use case ID	2
Samtidige forløb	1
Primær aktør	Brugeren
Sekundær aktør	Database
Initialisere	Brugeren
Mål	Systemet er opstartet
Forudsætninger	Use case 1 er gennemført.
Resultat	Systemet er nulpunktsjusteret og brugeren er klar til at blive forbundet til systemet
<hr/>	
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren indtaster login-oplysninger og trykker på "Log ind"-knappen. Systemet tjekker i databasen om oplysninger er gyldige [1.a Forkert login] 2. Brugeren trykker på "nulstil"-knappen. Systemet laver nulpunktsjustering
<hr/>	
Undtagelser	1a. Besked om forkert login vises. Use Case fortsættes fra punkt 1

Tabel 2.4: Fully dressed Use Case 2

Use Case 3

Navn	Mål blodtryk
Use case ID	3

Samtidige forløb	1
Primær aktør	Brugeren
Initialisere	Brugeren
Mål	Blodtryksmåling kører
Forudsætninger	UC2 er gennemført
Resultat	Blodtrykket vises i kontinuerlig graf, systolisk og diastoliske blodtryk vises grafisk, samt puls vises grafisk
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren trykker på "start"-knappen i blodtryksvindue 2. Transducere måler blodtryk 3. Systemet modtager blodtryksmåling fra transducere 4. Blodtryksgraf, systolisk, diastolisk og puls vises grafisk i blodtryksvinduet
Undtagelser	

Tabel 2.5: Fully dressed Use Case 3

Use Case 4

Navn	Filtrer signal
Use case ID	4
Samtidige forløb	2
Primær aktør	Brugeren
Initialisere	Brugeren
Mål	Brugeren ønsker at foretage en digital filtrering
Forudsætninger	UC2 er gennemført
Resultat	Det filtrerede signal vises i blodtryksgrafen
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren trykker på "Til"-knappen under filtrer 2. Systemet filtrerer signalet 3. Det filtrerede signal vises i blodtryks vindue
Undtagelser	

Tabel 2.6: Fully dressed Use Case 4

Use Case 5

Navn	Alarmer bruger
Use case ID	5
Samtidige forløb	2
Primær aktør	Brugeren
Initialisere	Brugeren
Mål	Systemet kan alarmere bruger ved for højt/lavt blodtryk
Forudsætninger	UC2 er gennemført
Resultat	Systemet alarmerer bruger
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren tilpasser diastoliske og systoliske grænseværdier ud fra patientens blodtryk 2. Systemet tjekker om grænseværdier er overskredet 3. Målte blodtryk overskrider valgte grænseværdier [3.a Målte blodtryk overskrider ikke valgte grænseværdier] 4. Alarm startes [4.a Bruger udskyder alarm]
Undtagelser	<ol style="list-style-type: none"> 3.a Use case fortsættes fra punkt 2 4.a Alarm udskydes med 3 minutter og use case starter fra punkt 2 igen

Tabel 2.7: Fully dressed Use Case 5

Use Case 6

Navn	Gem måling
Use case ID	6
Samtidige forløb	1
Primær aktør	Brugeren
Mål	Brugeren ønsker at afslutte systemet og gemme måling

Forudsætninger	UC3 er gennemført
Resultat	Blodtryksmålingens data er gemt i database og bruger er logget ud af systemet
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren trykker på "Gem og afslut"-knappen. "Gemme"-vindue åbnes. [1.a <i>Bruger ønsker ikke at afslutte</i>] 2. Brugeren indtaster CPR-nr og trykker på "Gem"knappen [2.a <i>CPR-nr er ikke gyldigt</i>] 3. Besked om at data er gemt vises. Brugeren trykker på "Ok"-knappen. Systemet logger ud og afsluttes
Undtagelser	<ol style="list-style-type: none"> 1.a. Bruger trykker på "Annuller"-knappen. Systemet lukker "Gemme"vinduet ned 2.a. Besked om at CPR-nummer ikke gyldigt, vises. Nyt CPR-nummer indtastet. Use Case fortsættes med punkt 2

Tabel 2.8: Fully dressed Use Case 6

2.4 Ikke-funktionelle krav

2.4.1 (F)URPS+

MoSCow er angivet i parentes ved hhv. M, S, C og/eller W, for Must, Should, Could og Won't

Functionality

1. (M) Brugeren skal kunne starte en ny måling indenfor XX sekunder efter opstart af programmet
2. (M) Systemet skal kunne kalibrere blodtrykssignalet
3. (M) Systemet skal kunne foretage en nulpunktsjustering
4. (M) Systemet skal kunne forstærke signalet fra transduceren (INDSÆT VÆRDI)
5. (M) Systemet skal kunne filtrere signalet med det indbyggede analoge antialiaseringsfilter med en båndbredde på 50 Hz

6. (M) Programmet skal kunne vise blodtrykket som funktion af tiden
7. (M) Programmet skal kunne vise blodtrykssignalet kontinuert
8. (M) Programmet skal programmeres i C#
9. (M) Programmet skal kunne lagre de målte data i en database
10. (M) Programmet skal kunne filtrere blodtrykket via et digitalt filter
11. (S) Programmet bør kunne afbilde både systolisk og diastolisk blodtryk med tal
12. (S) Programmet bør kunne måle puls
13. (C) Programmet kan angive pulsslag med bip-lyde med varighed af 100ms og en frekvens på 850 Hz

Usability

1. (M) Blodtrykstallene der udskrives på brugergrænsefladen er røde
2. (S) Pulsmålingen burde kunne udskrives på brugergrænsefladen med grønne tal
3. (M) Brugeren skal kunne starte en måling maksimalt 20 sekunder efter opstart
Knapper??
Billede af brugergrænsefladen indsættes

Reliability

1. (M) Systemet skal kunne køre uden fejl i et år
2. (M) Systemet skal have en "mean time to restore" på højest 24 timer

3. Systemet får herved en tilgængelighed beregnet ved

$$Availability = \frac{MTBF}{MTBF + MTTR} = \frac{365}{365 + 1} = 0,997 = 99,7\% \quad (2.1)$$

Performance

1. (S) Systemet bør kunne gemme data på 5 sekunder +/-10%

Supportability

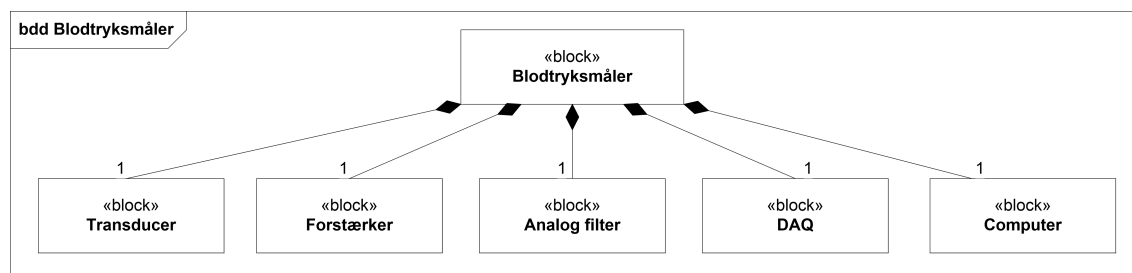
1. (M) Softwaren er opbygget af trelagsmodellen

Version	Dato	Ansvarlig	Beskrivelse
---------	------	-----------	-------------

3.1 Indledning

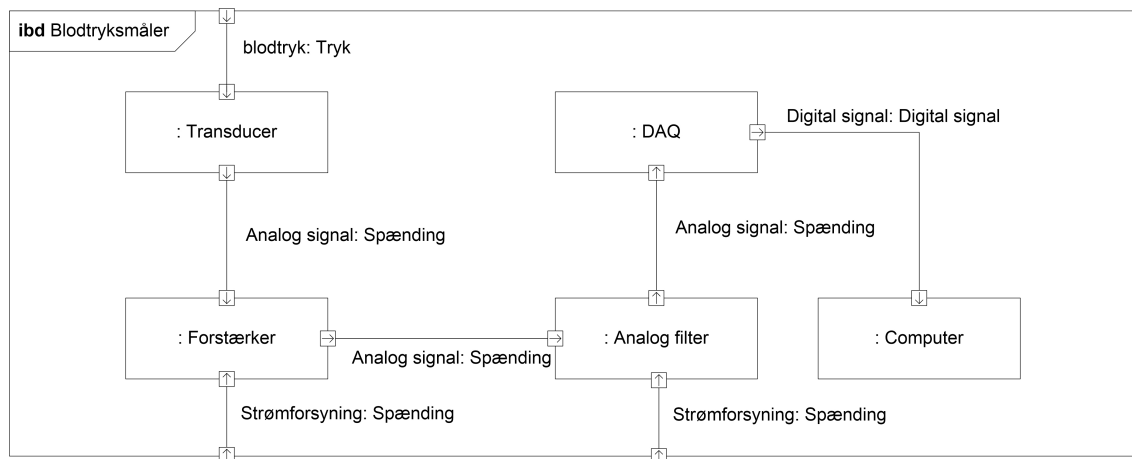
3.2 Hardware arkitektur

I følgende afsnit beskrives hvordan blodtryksmåler systemet og nogle af dets delkomponenter er opbygget.



Figur 3.1: Blokdiagram for blodtryksmåler systemet.

Ud af blokdiagrammet, figur 3.1, kan man se at blodtryksmåler systemet består af en transducer, en forstærker, et analogt filter, en DAQ og en computer.



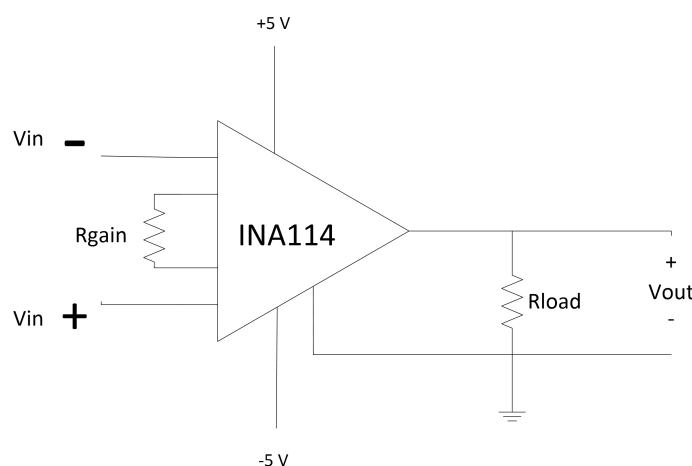
Figur 3.2: Internt blok diagram for blodtryksmåler systemet.

Ud af det interne blok diagram, figur 3.2, kan ses det at blodtrykket i form af det målte tryk kommer ind i transducere. Transducere som omformer det målte tryk til et spændingssignal, sender signalet videre til forstærkeren. Fra forstærkeren sendes signalet over i det analoge filter og derfra ind i DAQ'en. Endeligt sendes det digitale signal fra DAQ'en over i en computer, der fortolker signalet som et billede, der vises til omverdenen.

3.2.1 Design af forstærker

Forstærkeren er designet med tanke på, at det er meget små spændinger der arbejdes med. Grundet dette, er en almindelig operationsforstærker fravalgt, da dens reelle indgangsimpedans er for lille. En Instrumentationsforstærkers indgangsimpedans i den virkelige verden, er højere, og den kan dermed opfange meget små signaler så som blodtryk.

Vejleder rådede herefter til at projektgruppen brugte instrumentationsforstærkeren INA114. Forstærkerens design kan ses på figur X.



Figur 3.3: Der overordnede design af forstærkeren.

R_{gain} er modstanden, som bestemmer, hvor meget forstærkning, instrumentationsforstærkeren skal give og R_{load} er den belastning, der kommer efter kredsløbet. I dette tilfælde, er

det, det analoge filter. For at finde R_{gain} 's størrelse, kræver det, at der vides, hvor meget forstærkning der er brug for. Dette findes, ved at bestemme den maksimale spænding, som transduceren kan give, i en blodtrykssituation. Dette regnestykke kan ses realiseret i ligning 3.1:

$$VT_{max} = T_{max} \cdot V_{max} \cdot HG_{max} = \frac{\mu V}{V/mmHG} \cdot 5V \cdot 250mmHg = 6,25mV \quad (3.1)$$

Spændingen ønskes at skaleres op til DAQ'ens dynamikområde, som ligger omkring 2,5V. Forstærkningsfaktoren udregnes ved simpel brøkgregning, som ses på ligning 3.2:

$$G = \frac{2,5}{6,25 \cdot 10^{-3}} = 400 \quad (3.2)$$

INA114's datasheet giver en ligning for udregning af forstærkning. Da forstærkningen er kendt, omskrives ligning 3.3, så modstanden R_{gain} 's værdi i stedet bestemmes:

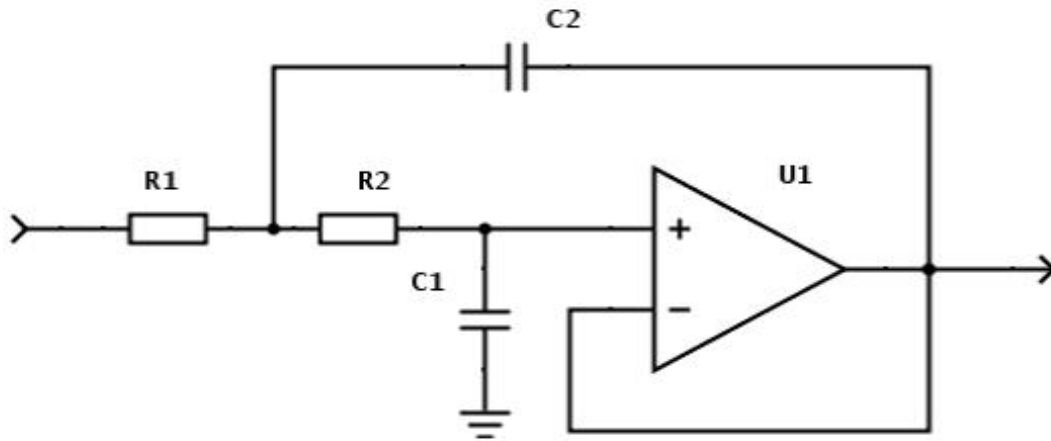
$$Gain = 1 + \frac{50k\Omega}{R_{gain}} \rightarrow G - 1 = \frac{50k\Omega}{R_{gain}} \rightarrow \frac{50k\Omega}{G - 1} = R_{gain} \quad (3.3)$$

Herefter kan den ohmske værdi af R_{gain} bestemmes, hvilket sker i ligning 3.4:

$$R_{gain} = \frac{50k\Omega}{400 - 1} = 125,31\Omega \quad (3.4)$$

3.2.2 Design af analogfilter

Filteret, som ses på figur 3.4, skulle realiseres som et aktivt 2. ordens lavpasfilter med båndbredde på 50 Hz af typen Sallen-Key med unity gain (se 3.4). Filteret skulle designes som et Butterworth filter med cut off frekvens på 50 Hz. C2 skulle vælges til 680 nF og $R1 = R2$. Operationsforstærkeren skal være af typen OP27.



Figur 3.4: Unity gain 2. ordens Sallen-Key lavpas konfiguration

Der blev valgt en dæmpningsfaktor på 1, da det ville være mest optimalt med et kritisk dæmpet signal. Ud fra hjemmesiden¹ fandt vi overføringsfunktionen for Sallen-Key lavpasfiltret, udtrykt i ligning 3.5:

$$\frac{V_{out}(S)}{V_{in}(S)} = \frac{\frac{1}{R_1 C_1 R_2 C_2}}{s^2 + s\left(\frac{1}{R_2 C_2} + \frac{1}{R_1 C_2}\right) + \frac{1}{R_1 C_1 R_2 C_2}} \quad (3.5)$$

Da det er blevet opgivet at $R_1 = R_2$, kan overføringsfunktionen forkortes:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{1}{R^2 C_1 C_2}}{s^2 + s\left(\frac{2}{RC_2}\right) + \frac{1}{R^2 C_1 C_2}} \quad (3.6)$$

Dernæst sammenlignes med standardformlen for overføringsfunktionen for et andet ordens filter.

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{1}{R^2 C_1 C_2}}{s^2 + s\left(\frac{2}{RC_2}\right) + \frac{1}{R^2 C_1 C_2}} = \frac{\omega_0^2}{s^2 + s(2\zeta\omega_0) + \omega_0^2} \quad (3.7)$$

Ud fra dette kan regnes komponentværdierne for R, idet

$$\frac{2}{RC_2} = 2\zeta\omega_0 \quad (3.8)$$

Ved hjælp af mathcad isoleres R.

¹<http://sim.okawa-denshi.jp/en/OPseikiLowkeisan.htm>

$$\frac{2}{R \cdot 680 \cdot 10^{-9}} = 2 \cdot 1 \cdot (50 \cdot 2 \cdot \pi) \text{ solve, } R \rightarrow \frac{250000}{17 \cdot \pi} = 4.681 \times 10^3 \quad (3.9)$$

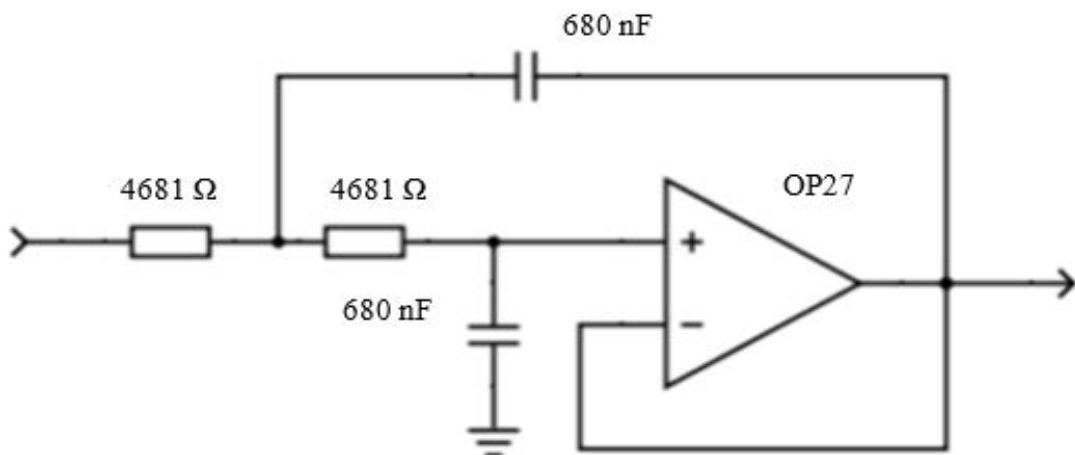
Dernæst kan komponentværdien for C1 udregnes, idet:

$$\frac{1}{R^2 C_1 C_2} = \omega_0^2$$

Ved hjælp af mathcad isoleres C1.

$$\frac{1}{(4.681 \times 10^3)^2 \cdot (680 \cdot 10^{-9}) \cdot C_1} = (50 \cdot 2\pi) \text{ solve, } R \rightarrow \frac{250000}{17\pi} = 4.681 \times 10^3 \quad (3.10)$$

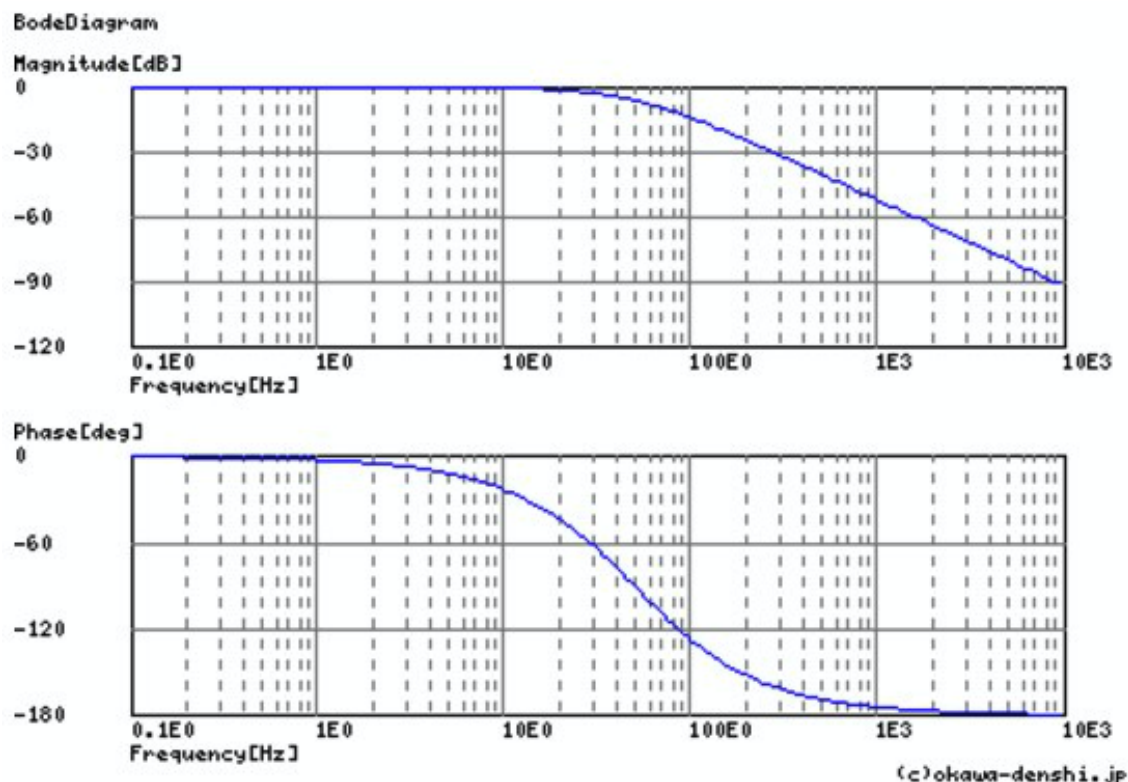
Derved er komponentværdierne for kredsløbet fundet og de ses indskrevet på 3.4.



Figur 3.5: Unity gain 2. ordens Sallen-Key lavpas konfiguration med indsatte komponentværdier.

Generelt er et unity gain Sallen-Key filter med equivalente capacitorer og equivalente resistore kritisk dæmpet dvs. en kvalitets faktor på 1/2. Dette kan også ses når komponentværdierne indsættes i "Sallen-Key Low-pass Filter Design Tool"². Desuden ses bodeplottet nedenfor:

²<http://sim.okawa-denshi.jp/en/OPseikiLowkeisan.htm>



Figur 3.6: Bodeplot af overføringsfunktionen³

3.2.3 Grænseflader

3.3 Software arkitektur

3.3.1 GUI

Dette afsnit beskriver hvilke tanker og overvejelser der er gjort i forbindelse med designet af diverse brugergrænseflader. Nedenfor ses udkast til disse brugergrænseflader.

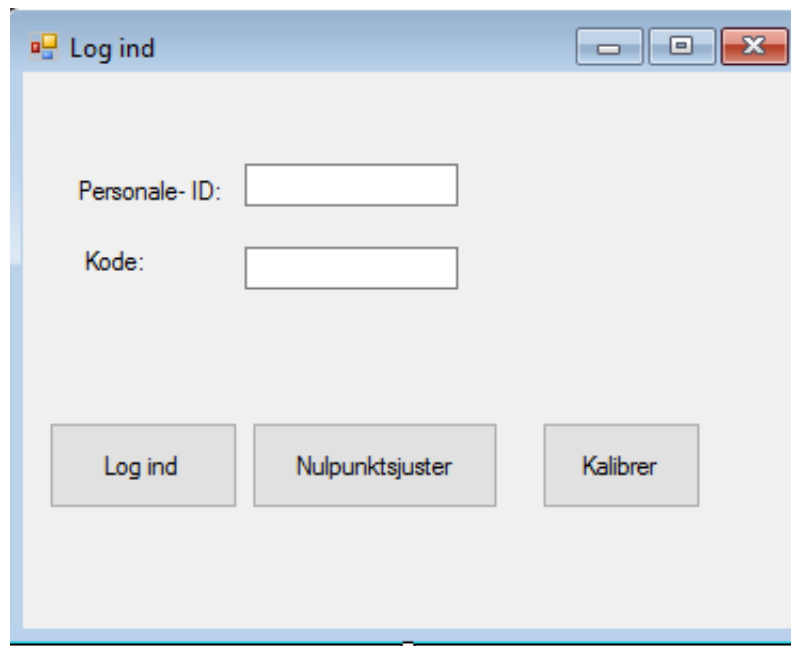
Til designet af brugergrænsefladen er blevet udført ud fra de 16 principper⁴ for gode brugergrænseflader. Der er ligeledes i form1 hentet inspiration fra allerede eksisterende blodtryk monitor. I designet af brugergrænsefladen bestræbes, der efter at gøre det så virkelighedsnært som muligt ud fra de redskaber og oplysninger der er til rådighed.

Brugergrænsefladen er inddelt i 3 forskellige interfaces. De 3 interfaces er henholdsvis Log ind, Form1 og Gem. De personer, der interagerer med brugergrænsefladen er sundhedsfagligt personale som typisk vil være en læge eller sygeplejerske. Dette er der også taget højde for i designet da, der aldrig vil være andre end det sundhedsfaglige personale der interagerer med brugergrænsefladen og derfor er den en skærpet målgruppe.

Nedenfor kommer der en uddybende beskrivelse af brugergrænsefladen.

³ <http://sim.okawa-denshi.jp/en/OPseikiLowkeisan.htm>

⁴ https://en.wikibooks.org/wiki/Usability_for_Nerds



Figur 3.7: Login vindue

I figur 3.7 er der taget udgangspunkt i, at brugergrænsefalden skal være enkel og overskuelig opbygget. Der skal ikke være nogle overflødige ting og de knapper der er vigtige som bruger skal benytte skal stå tydeligt frem. Opbygningen skal afspejle brugerens logik.

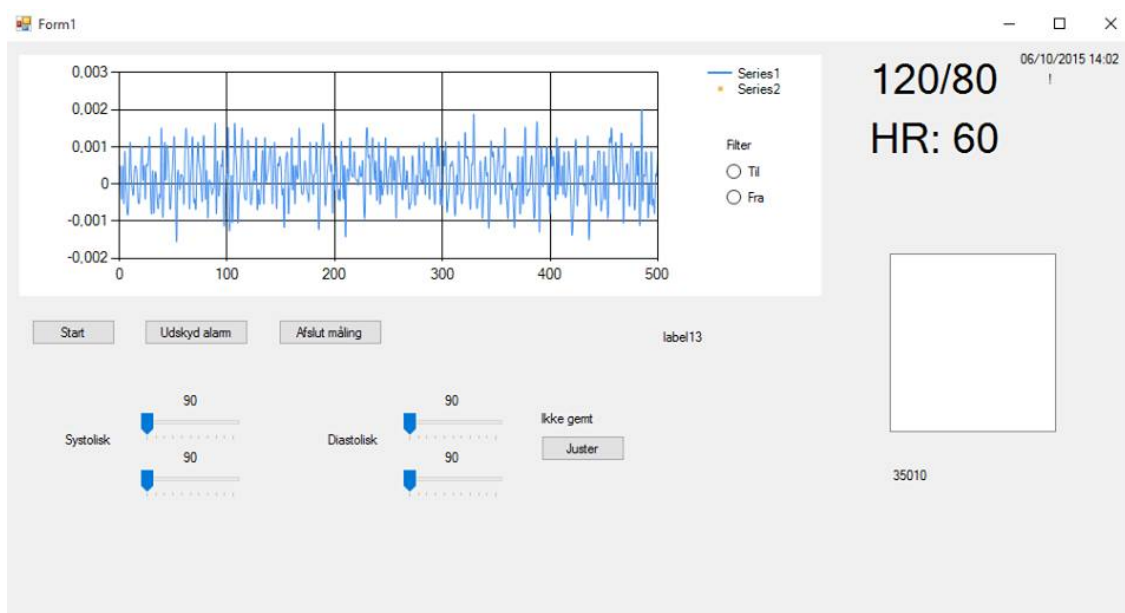
Tankerne omkring den enkle opbygning er, at hvis der opstår en akut situation hvor systemet skal i gang hurtigt skal det være nemt og hurtigt at logge ind i systemet. Det er vigtigt at knapperne er selvforklarende og giver brugeren det der forventes af knappen når den benyttes.

Der er taget højde for feedback til bruger ved forkert log ind får brugeren en besked om at log ind er indtastet forkert og der skal prøves forfra.



Figur 3.8: Kalibrer vindue

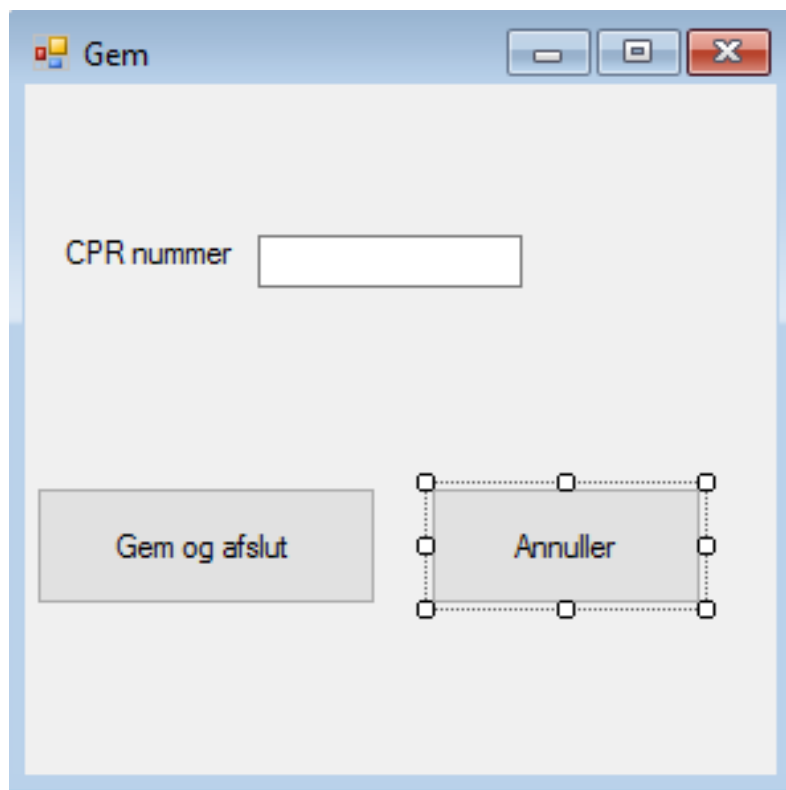
Vist på figur 3.8, hvor der her er lagt vægt på det funktionelle, med et enkelt og simpelt design. Der er vejledende tekst, som gør det nemt for aktøren at finde ud af systemet.



Figur 3.9: Blodtryk vindue

I blodtryksvinduet, vist på figur 3.9, er opbygningen mere kompliceret og med flere knapper end ved Log ind og Gem. Der er taget højde for at knapperne er selvforklarende og deres navne afspejler de bagvedliggende handlinger. Dette medfører at brugeren stadig har et

overblik over brugergrænsefalden og stadig selv kan kontrollere hvad der skal ske. Denne brugergrænseflade er ikke tiltænkt nye brugere, men derimod brugere der har et kendskab til den og til hvilke sundhedfaglige værdier og udtryk, der vises på den. Derfor benyttes brugerens sprog på brugergrænsefalden og ikke et sprog med termer, der er forståelig for alle.



Figur 3.10: Gem vindue

På figur 3.10, gemme-vinduet, er der igen taget udgangspunkt i en enkel opbygning og at knapperne, der skal bruges er tydelige og selvforklarende. Det er her også brugerens logik, der afspejles og ingen unødvendige ting er inddraget i brugergrænsefalden. Det er her muligt at afbryde situationen ved at der er en alternativ udvej, som kan bruges hvis den op startet handling fortrydes. Der er taget højde for feedback til brugeren hvis der indtastes forkert CPR nummer får brugeren det af vide i et pop-up vindue.

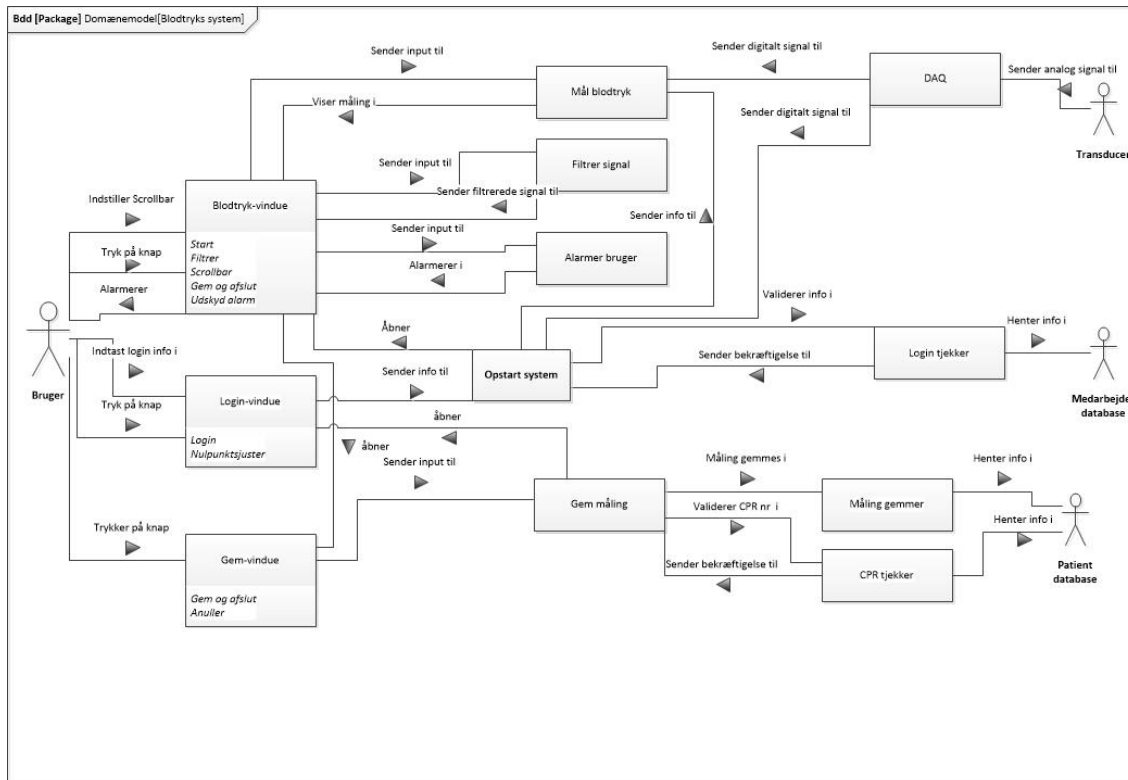
3.3.2 UML klassediagram

3.3.3 Applikationsmodel

Klassediagram

Domænemodel

Diagrammet viser en domæne model for blodtryks systemet. Dette diagram giver et godt overblik over systemet som helhed og hvilke elementer, der indgår i systemet. Diagrammet viser hvordan brugeren interagerer med systemet og hvordan systemet forløber efter det igangsættes af bruger.

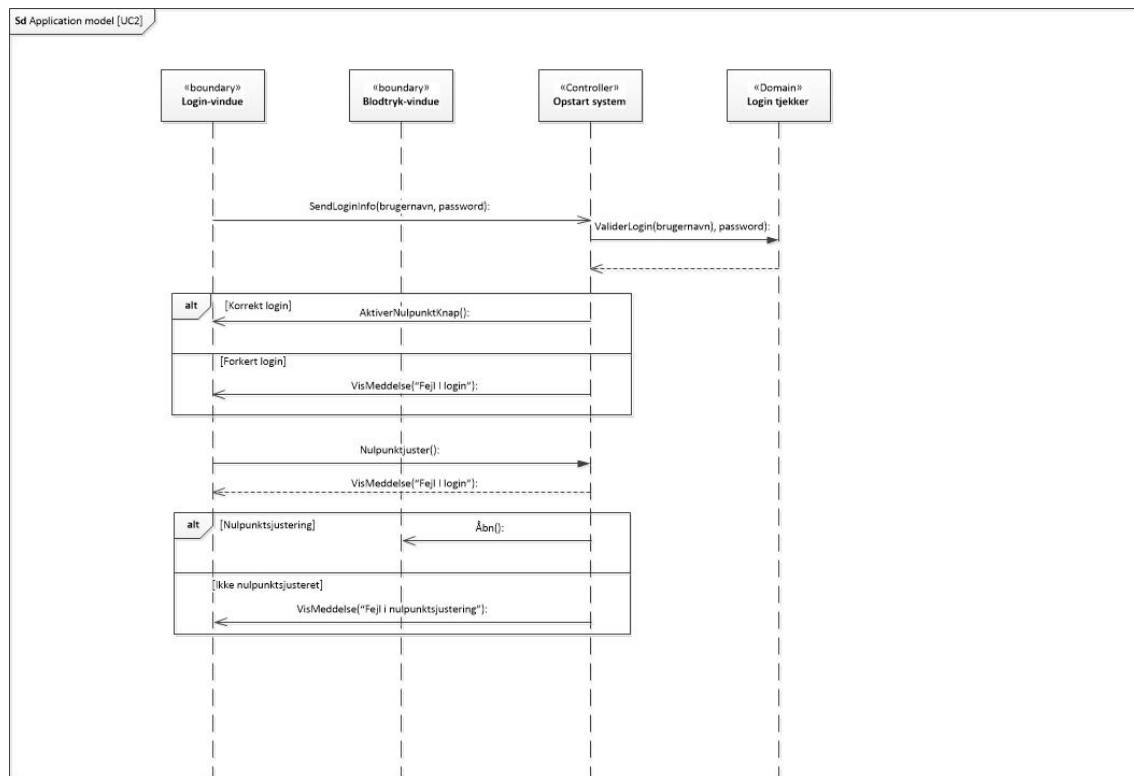


Figur 3.11: Domænemodel

Sekvensdiagram

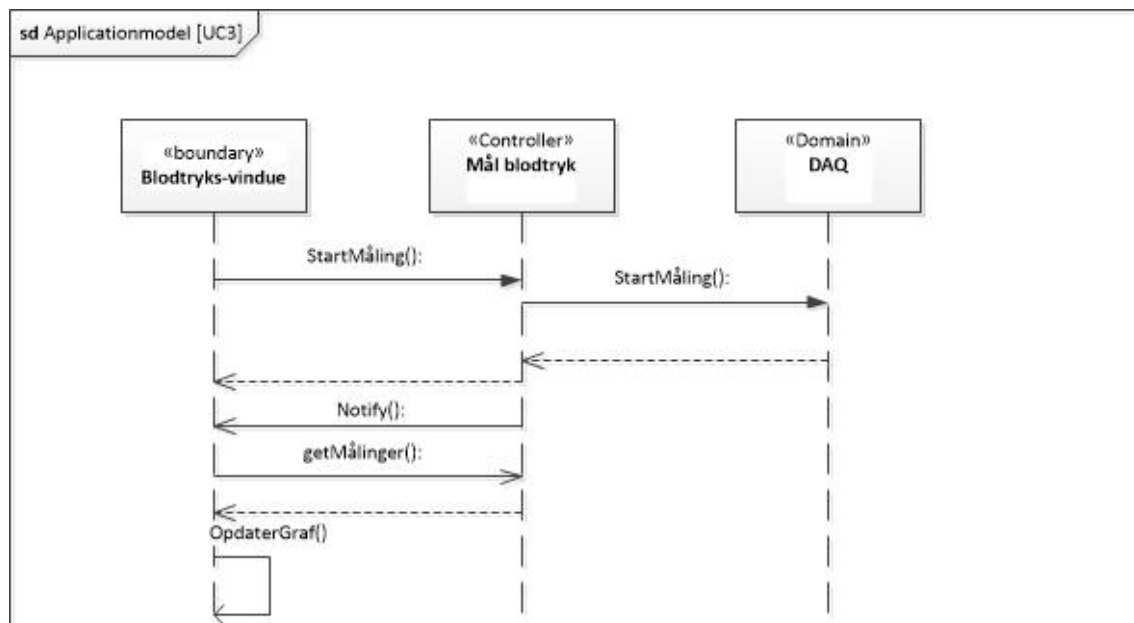
Sekvensdiagrammet er et interaktions diagram, der viser hvordan processer i systemet forløber. Use casene ligger til baggrund for udarbejdelsen af sekvensdiagrammerne.

Der er blevet lavet et sekvensdiagram for hver use case for at give det bedste overblik over hver handling i forhold til systemet. I sekvensdiagrammet har vi brugt virtuelle metode kald til at beskrive forløbet. I use case 2-7 er det brugeren, der interagerer med systemet og er initiator for a handlingerne bliver udført.



Figur 3.12: Sekvensdiagram UC2

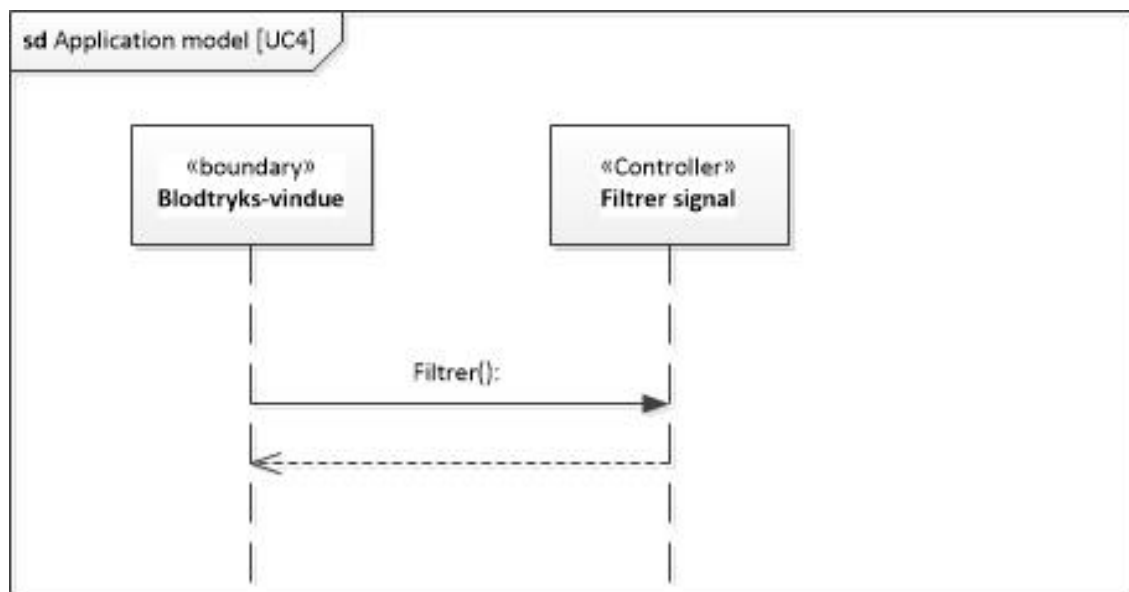
På figur 3.12 ses det hvordan brugeren logger ind i systemet og hvordan brugeren for systemet nulpunktsjusteret. Det er brugeren, der interagerer med systemet og er initiator for at handlingerne bliver udført.



Figur 3.13: Sekvensdiagram UC3

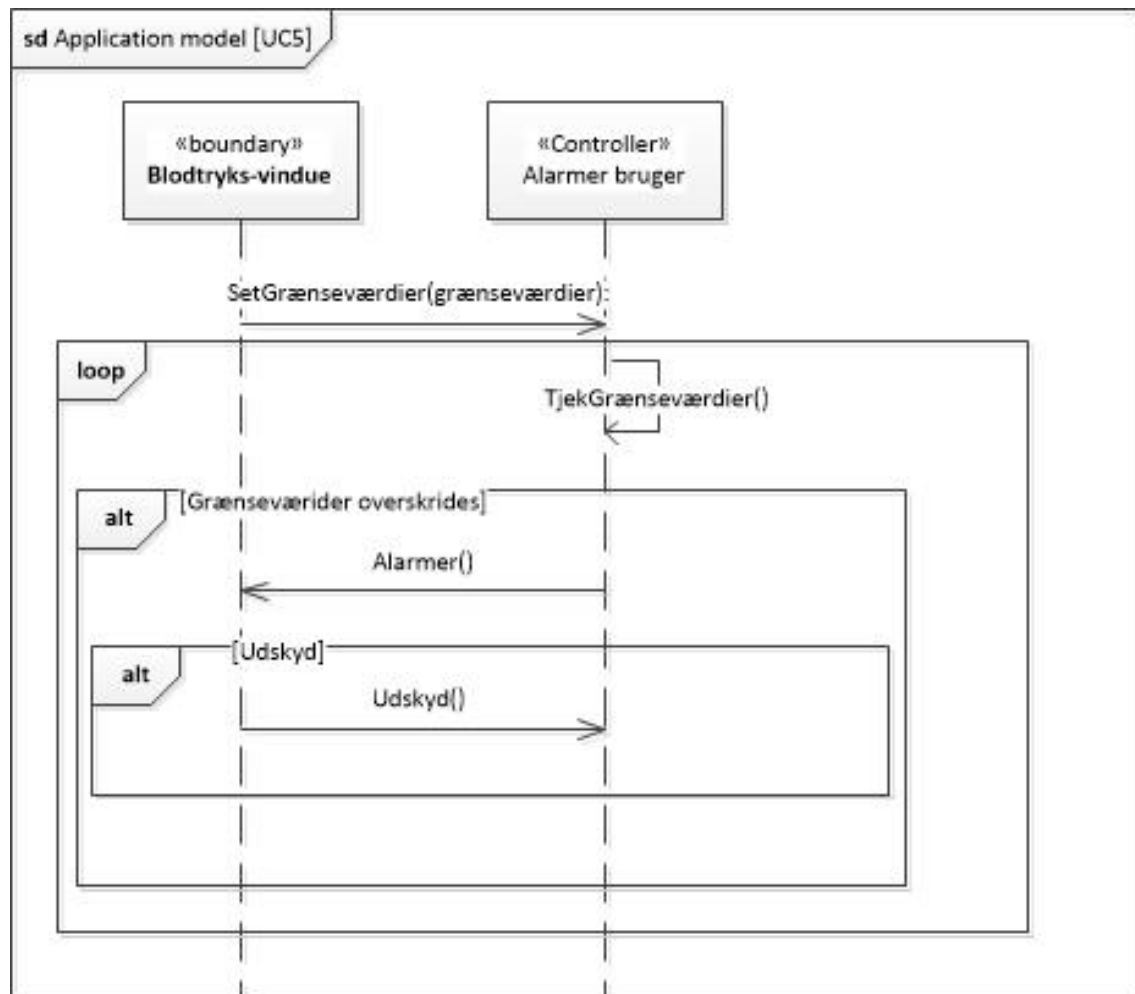
Figur 3.13 viser hvordan brugeren starter målingen og hvordan graferne vises på

brugergrænsefalden.



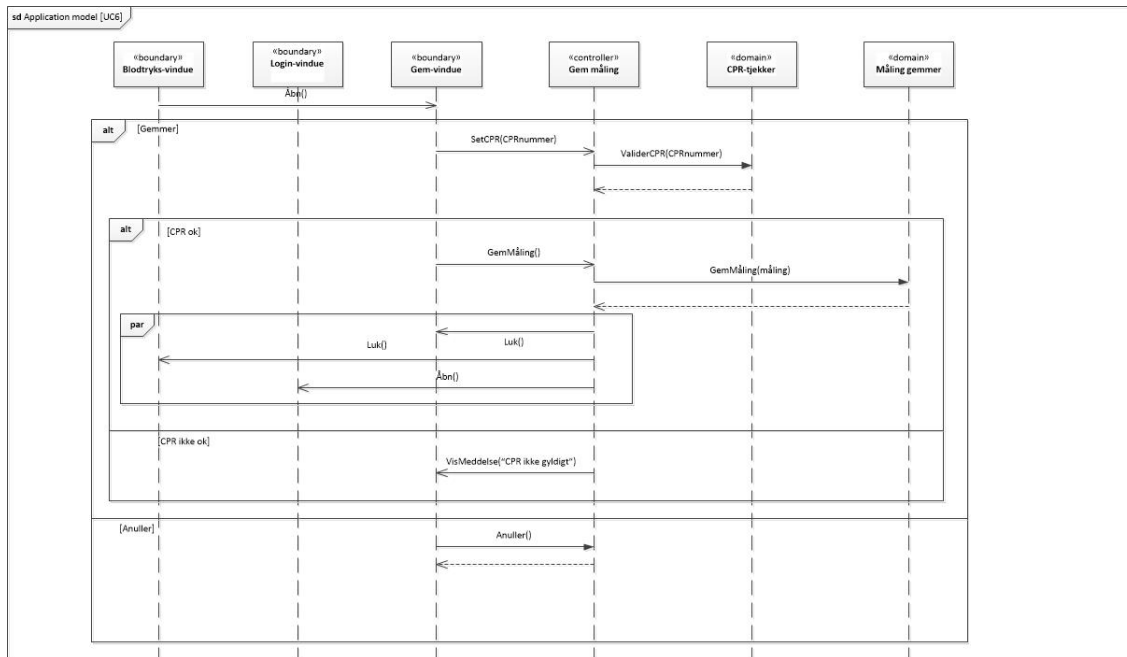
Figur 3.14: Sekvensdiagram UC4

I sekvensdiagrammet figur 3.14 ses hvordan brugeren kan vælge om blodtrykssignalet skal filtreres eller ej.



Figur 3.15: Sekvensdiagram UC5

Det ses på figur 3.15 hvordan brugeren justerer grænseværdierne for patientens blodtryk. Denne justering sker ud fra patientens målte blodtryk. De justerede grænseværdi ligger til grundlag for alarmen. Desuden viser figuren, at brugeren har mulighed for at udskyde alarmen.

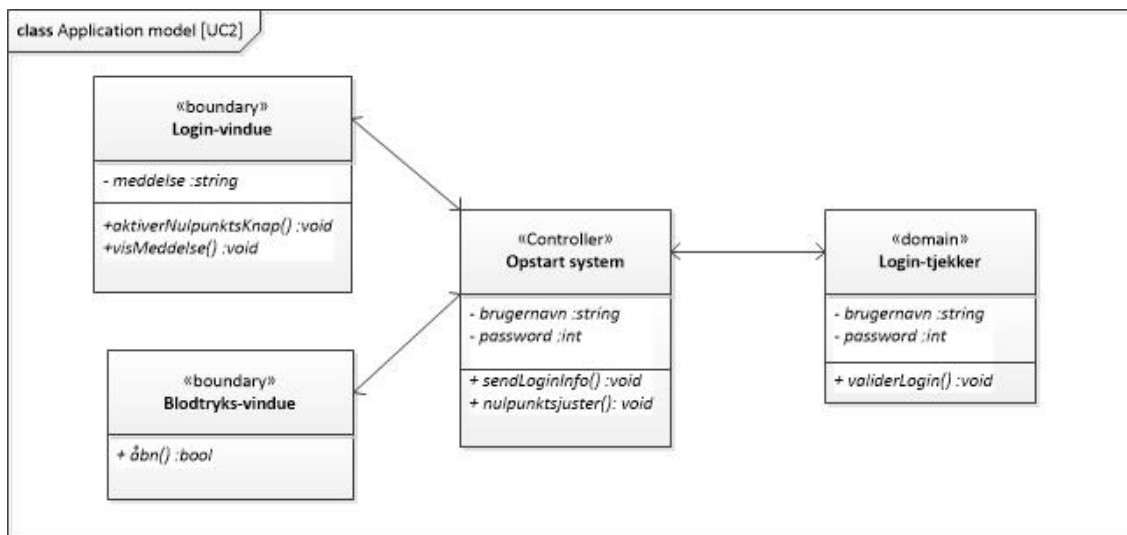


Figur 3.16: Sekvensdiagram UC6

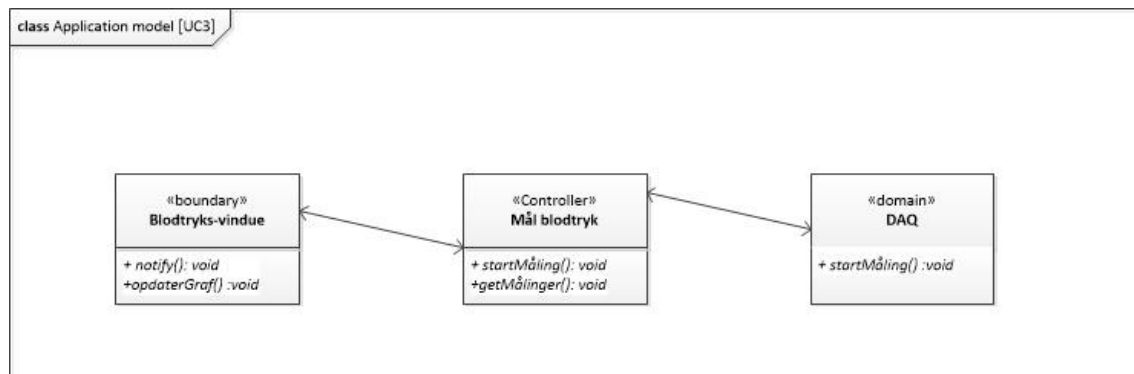
Figur 3.16 viser hvordan systemet gemmer og afslutter en måling.

Opdateret Klassediagram

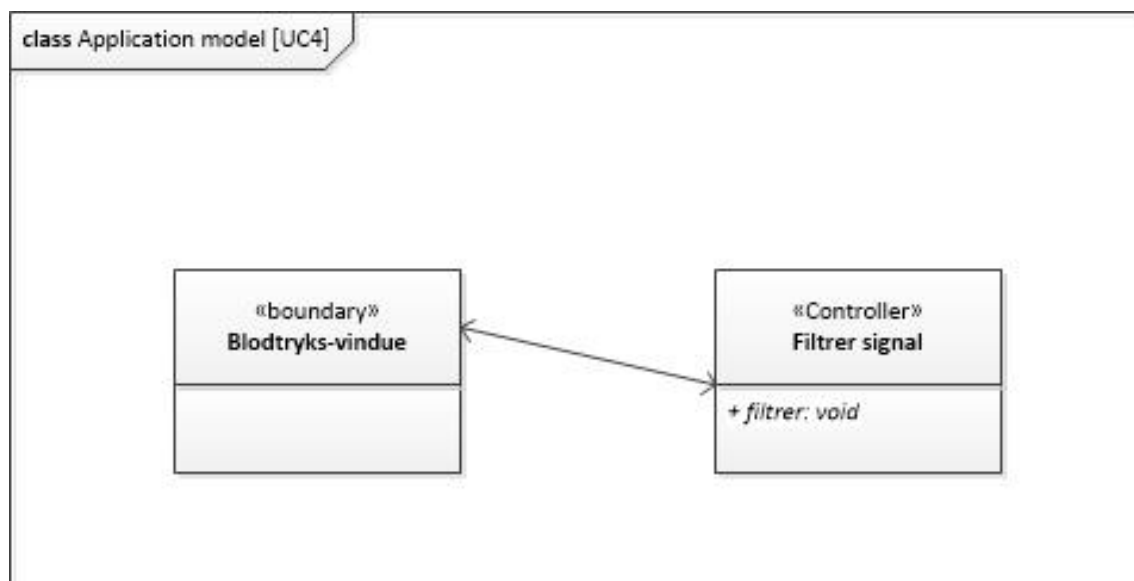
Klasse applikationsmodellerne er udarbejdet ud fra use casene, sekvensdiagrammer og domænemodellen. Der er ligesom ved sekvensdiagrammerne lavet et klassediagram for hver use case. Der er her også brugt virtuelle metoder.



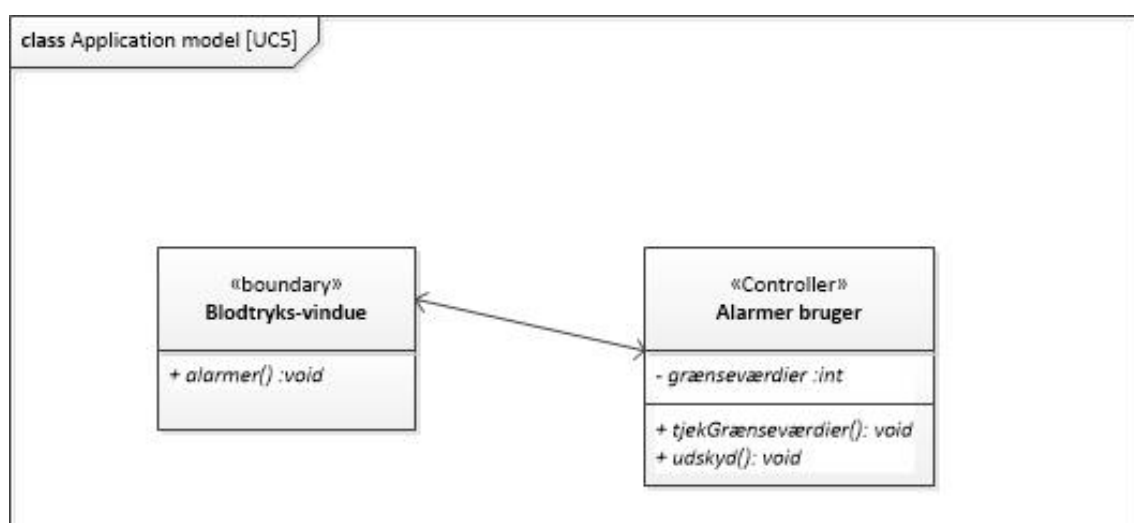
Figur 3.17: Klassediagram UC 2



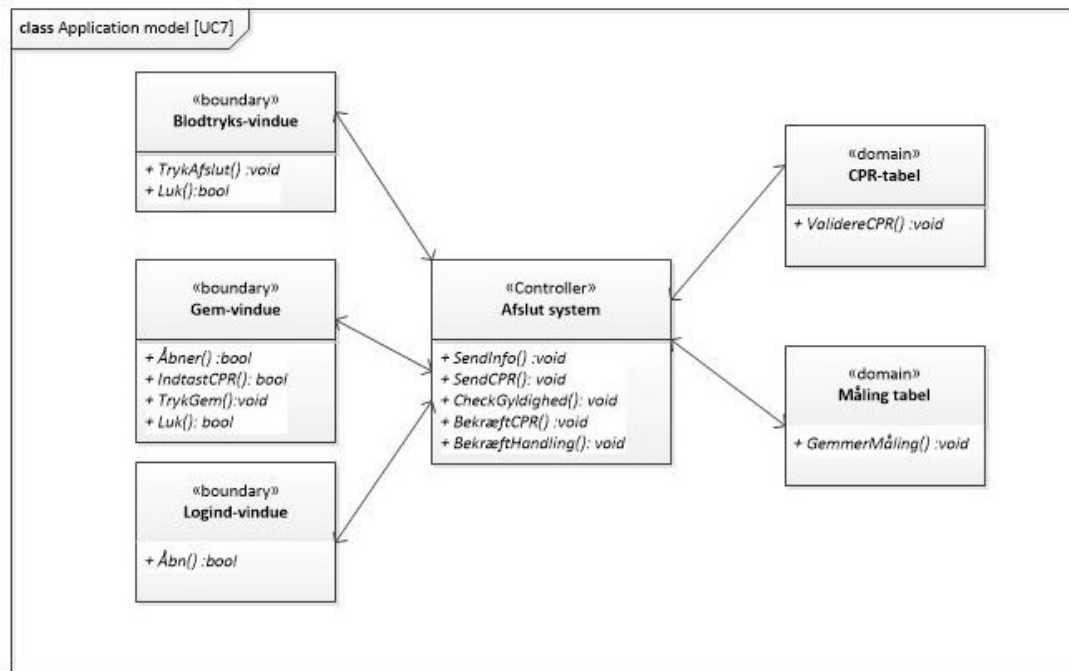
Figur 3.18: Klassesdiagram UC 3



Figur 3.19: Klassesdiagram UC 4



Figur 3.20: Klassesdiagram UC 5



Figur 3.21: Klassesdiagram UC 7

3.4 Software implementering

Integrationstest 4

Version	Dato	Ansvarlig	Beskrivelse
---------	------	-----------	-------------

4.1 Hardware

4.2 Software

Accepttest 5

Version	Dato	Ansvarlig	Beskrivelse
---------	------	-----------	-------------

5.1 Indledning

I acceptttesten testes de krav, der er opstillet i kravspecifikationen. Det fremgår af acceptttesten om hvilke scenarier og krav, der er blevet opfyldt og implementeret i systemet og hvilke, der enten ikke er testbare eller ikke er blevet implementeret.

Først vil der være acceptttest af use casene altså de funktionelle krav efterfulgt af acceptttest af de ikke funktionelle krav.

5.2 Accepttest af Use Cases

5.2.1 Use Case 1

Kalibrer System

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Forbind systemet til en kendt trykkilde	Systemet er forbundet til en kendt kilde		
2. Tryk 'Start'	Et tryk der kan differentiere fra det kendte kommer frem på skærmen		
3. Skriv resultatet ned og gentag punkt 2., 2 gange mere	Der er tre måleafvigelses skrevet ned		
4. Udregn gennemsnit for afvigelse og omregn til procenter	Afvigelses koefficient er fundet		
5. Indtast afvigelses koefficient i tekstboksen	Systemet er kalibreret		

- | | | |
|----|---------------------|---------------------------|
| 1. | Systemet kalibreres | At systemet er kalibreret |
|----|---------------------|---------------------------|

Tabel 5.2: Accepttest af Use Case 1.

5.2.2 Use Case 2

Opstart system

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1.	Indtast personale-ID i brugernavnsfeltet; "1234"og indtast personlig kode i kodeordsfeltet; "fido"	Loginoplysninger bliver udfyldt	
2.	Tryk på "Log ind"-knappen	Log ind oplysninger er gyldige og stemmer overens med hinanden. Tekstfelter til log ind skjules og "Nulstil"-knappen vises	
3.	Tryk på "Nulstil"-knap	Besked om at systemet er nulpunktsjusteret vises i "Ok"-vinduet som åbnes	
4.	Tryk på "Ok"-knappen	"Log ind"-vinduet og "Ok"- vinduet lukkes og "Blodtryk"-vinduet åbnes	
2.a	Log ind oplysninger findes ikke i databasen. Besked vises med tekst, der informerer herom	Nye log ind oplysninger indtastes	

Tabel 5.3: Accepttest af Use Case 2.

5.2.3 Use Case 3

Mål blodtryk

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Tryk på "start"-knappen i i blodtryksvindueet	Graf og blodtryks værdier vises på brugergrænsefladen		

Tabel 5.4: Accepttest af Use Case 3.

5.2.4 Use Case 4

Filtrer signal

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Påsæt sinus signal med frekvens XXHz(højfrekvent)	Sinus signal vises på grafen		
2. Tryk på "Til"-knappen under filter	Signalet udglattes		
3. Påsæt sinus signal med frekvens XXHz(lavfrekvent)	Sinus signal vises på grafen		
4. Tryk på "Fra"-knappen under filter	Sinus-signal udglattes ikke		

Tabel 5.5: Accepttest af Use Case 4.

5.2.5 Use Case 5

Juster grænseværdier

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			

1.	Indtast 100 for diastolisk øvre grænse	Der står 100 i pågældende tekst felt
2.	Indtast 80 for diastolisk nedre grænse	Der står 100 i pågældende tekst felt
3.	Indtast 100 for systolisk øvre grænse	Der står 120 i pågældende tekst felt
4.	Indtast 80 for systolisk øvre grænse	Der står 80 i pågældende tekst felt
5.	Tryk på 'Start' knappen i blodtryksvindue	Blodtryksmåling startes og Alarm går igang

Tabel 5.6: Accepttest af Use Case 5.

5.2.6 Use Case 6

Afslut system

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1.	Tryk på "Afslut"-knappen	"Gemme"-vinduet vises	
2.	Indtast CPR-nr "1111111111"	CPR-nummeret synligt i pågældende tekst felt	
3.	Tryk på "Gem og afslut"-knappen	"Gemme"-vindue og "Blodtryks"-vinduet lukkes. "Login"-vinduet vises?	
<i>Undtagelser</i>			
1.a.	Tryk på "annuller"-knap	"Gemme"-vinduet lukkes og "Blodtryk"-vinduet vises	
2.a.	Indtast CPR-nummeret "1234567890"	Beskeden "CPR ikke gyldigt." vises	

Tabel 5.7: Accepttest af Use Case 7.

5.3 Accepttest af ikke-funktionelle krav

	Ikke-funktionelt krav	Test/handling	Forventet resultat	Faktiske observationer	Godkendt
<i>Functionality</i>					
1.	Brugeren skal kunne starte en ny måling indenfor XX sekunder efter opstart	Start programmet, hvorefter der vha. stopur måles opstartstiden	At programmet er opstartet og ny måling er igang efter XX sekunder		
2.	Systemet skal kunne kalibrere blodtrykssignalet	Opstart programmet, til den automatiske kalibrering er fuldført	At systemet har kaliberet signalet		
3.	Systemet skal kunne foretage en nulpunktsjustering	Tryk på "nulstil"-knap	At blodtryksgrafen bliver nulpunktsjusteret		
4.	Systemet skal kunne forstærke signalet med det indbyggede analoge antialiaseringsfilter med en båndbredde på 50 Hz	Start systemet	At signalet er forstærket		
5.	Programmet skal kunne vise blodtrykket som funktion af tiden	Tryk på "Start måling"-knap	At blodtrykket er vist som funktion af tiden på brugergrænsefladen		

6.	Programmet skal kunne vise blodtrykssignalet kontinuerligt	Tryk på "Start måling"-knap	At blodtryks-signalet er vist kontinuert på brugergrænsefladen
7.	Programmet skal programmeres i C#	Start programmet	At koden er i C#
8.	Programmet skal kunne lagre de målte data i en database	Tryk på "Gem"-knap	At målingen er gemt i database
9.	Programmet skal kunne filtrere blodtrykket via et digitalt filter	Tryk på "Filtrer signal" til på radiobutton	At det viste blodtrykssignal er filtreret
10.	I programmet skal det digitale filter kunne slås til og fra på en radiobutton	Tryk "Filtrer signal" til og fra på radiobutton	
11.	Programmet bør kunne afbilde både systolisk og diastolisk blodtryk med tal	Tryk "Start måling"	At systolisk og diastolisk blodtryk er afbilledet med tal på brugergrænseflade
12.	Programmet bør kunne måle og afbillede puls	Tryk på "Start måling"	At pulsen er afbilledet på brugergrænseflade
13.	Programmet bør kunne give alarm, hvis det systoliske blodtryk overstiger 140 mmHg eller falder under 100 mmHg	Påsat blodtryksignal fra Physionet der overskrider valgte grænser	At en alarm med frekvens på XX begynder

14.	Programmet bør kunne give alarm, hvis det diastoliske blodtryk overstiger 90 mmHg eller falder under XX mmHg	Påsat blodtryksignal fra Physionet der overskrider valgte grænser	At en alarm med frekvens på XX begynder
-----	--	---	---

15.	Programmet kan angive pulsslag med bib-lyde med varighed af 100 ms og en frekvens på 850 Hz	Tryk på "Start måling"	At pulsen høres som bib-lyde af 100 ms varighed og med en frekvens på 850 Hz
-----	---	------------------------	--

Usability

1.	Blodtrykstillene der udskrives på brugergrænseflade er røde	Tryk "Start måling"-knap	At Blodtrykstallene er røde
----	---	--------------------------	-----------------------------

2.	Pulsmålingen skal udskrives på brugergrænsefladen med grønne tal	Tryk "Start måling"-knap	At pulsen vises med grønne tal
----	--	--------------------------	--------------------------------

3.	Brugeren skal kunne starte en måling på maksimalt 20 sekunder	Start systemet op, log ind og tryk "Start måling"	At programmet er opstartet og ny måling er startet på under 20 sekunder
----	---	---	---

4.	Brugergrænsefladelever op til nedensående figur	Opstart program og log ind	At brugergrænseflade indeholder samtlige funktioner som på figuren
----	---	----------------------------	--

Reliability

1.	Systemet skal kunne køre uden fejl i et år	Start system op og vent et år	At programmet efter et år kører fejlfrit
2.	Systemet skal have en "mean time to restore" på højst 24 timer	Start systemet og herefter genstart, hvor der tages tid med et stopur	At programmet er klar igen inden for 24 timer
<i>Performance</i>			
1.	Systemet bør kunne gemme data på 5 sekunder +/- 10 sekunder	Tryk på "Gem og afslut"-knap og tag tid med stopur	At data er inden for 5 sekunder
<i>Supportability</i>			
1.	Softwareen er opbygget af tre-lagsmodellen	Kig i koden efter data-lag, logik-lag og GUI-lag	At koden indeholder et data-lag, et logik-lag og et GUI-lag

Tabel 5.8: Accepttest af Ikke-funktionelle krav

Fejlrapport

Logbog

Mødereferat

Kode

Samarbejdsaftale