



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

12. 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
2.3.2 Use case 1	5
2.3.3 Use case 2	6
2.3.4 Use case 3	7
2.3.5 Use case 4	7
2.3.6 Use case 5	8
2.3.7 Use case 6	8
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	19
3.3.1 GUI	19
3.3.2 Domænemodel	22
3.3.3 Applikationsmodel	23
Kapitel 4 Implementering	31
4.1 Hardware implementering	31
4.2 Software implementering	34
4.2.1 3-lagsmodellen	34
4.2.2 Tråde	38
4.2.3 Observer	38
4.2.4 Push pull	38
4.2.5 UML klassediagram	38
Kapitel 5 Test	39

5.1	Indledning	39
5.2	Modul test hardware	39
5.2.1	Modul test af forstærkeren	39
5.2.2	Modul test af det analoge filter	40
5.3	Modul test software	42
5.4	Integrationstest hardware	43
5.5	Integrationstest software	43
5.6	Integrationstest system	43
Kapitel 6	Accepttest	45
6.1	Indledning	45
6.2	Accepttest af Use Cases	45
6.2.1	Use Case 1	45
6.2.2	Use Case 2	46
6.2.3	Use Case 3	47
6.2.4	Use Case 4	48
6.2.5	Use Case 5	48
6.2.6	Use Case 6	49
6.3	Accepttest af ikke-funktionelle krav	50
Bilag		53
	Fejlrappport	53
	Logbog	53
	Mødereférerart	53
	Kode	53
	Samarbejds aftale	53

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

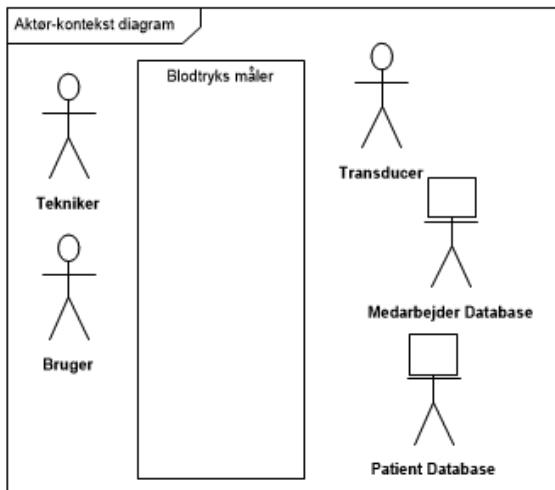
2.1 Indledning

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 analysere signalet samt viser signalet grafisk på en brugergrænseafdeling.

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 systemets 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, anestesi 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 tranduceren, der leverer blodtryksmålingen til systemet
Medarbejder database	Sekundær	Medarbejder databasen er det sted, hvor medarbejderens log ind valideres
Pateint 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

2.3.2 Use case 1

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	<ol style="list-style-type: none"> 1. Tekniker mäter och antecknar atmosfärisk tryck 2. Tekniker trycker på systemet tre kända tryck 3. Tekniker läser svarerna 4. Tekniker antecknar avvikelser från de kända tryck [3.a <i>Det är ingen avvikelser</i>] 5. Tekniker kalibrerar avvikelserna
Undtagelser	4.a Use case avslutas

Tabel 2.3: Fully dressed Use Case 1

2.3.3 Use case 2

Use Case 2

Navn	Opstart system
Use case ID	2
Samtidige forløb	1
Primær aktör	Brugeren
Sekundær aktör	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
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren indstiller transduceren til at mäta det atmosfäriske tryck 2. Brugeren trycker på "nulstil"-knappen. Systemet laver nulpunkts justering 3. Brugeren indtaster login-oplysningar och trycker på "Log in"-knappen. Systemet tjekker i databasen om oplysningar er gyldige [3.a <i>Forkert login</i>]
Undtagelser	3a. Besked om forkert login vises. Use Case fortsættes fra punkt 3

Tabel 2.4: Fully dressed Use Case 2

2.3.4 Use case 3

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. Transduceren måler blodtryk 3. Systemet modtager blodtryksmåling fra transduceren 4. Blodtrykgraf, systolisk, diastolisk og puls vises grafisk i blodtryksvinduet
Undtagelser	

Tabel 2.5: Fully dressed Use Case 3

2.3.5 Use case 4

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
1. Brugerens trykker på "Til"-knappen under filter
 2. Systemet filtrerer signalet
 3. Det filtrerede signal vises i blodtryks vindue
-

Undtagelser

Tabel 2.6: Fully dressed Use Case 4

2.3.6 Use case 5

Use Case 5

Navn	Alarmer bruger
Use case ID	5
Samtidige forløb	2
Primær aktør	Bruger
Initialisere	Bruger
Mål	Systemet kan alarmere bruger ved for højt/lavt blodtryk
Forudsætninger	UC3 er gennemført
Resultat	Systemet alarmerer bruger
Hovedforløb	<ol style="list-style-type: none"> 1. Bruger tilpasser diastoliske og systoliske grænseværdier ud fra patientens blodtryk 2. Systemet tjekker om grænseværdier er overskredet 3. Målte blodtryk overskridt valgte grænseværdier [3.a <i>Målte blodtryk overskridt ikke valgte grænseværdier</i>] 4. Alarm startes [4.a <i>Bruger udskyder alarm</i>]
Undtagelser	<p>3.a Use case fortsættes fra punkt 2</p> <p>4.a Alarm udskydes med 3 minutter og use case starter fra punkt 2 igen</p>

Tabel 2.7: Fully dressed Use Case 5

2.3.7 Use case 6

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 afslutter
Undtagelser	<ol style="list-style-type: none"> 1.a. Bruger trykker på "Annuler"-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 forstærke signalet fra transduceren 400 gange
3. (M) Systemet skal kunne filtrere signalet med det indbyggede analoge antialiaserings filter med en båndbredde på 50 Hz, samt en cut-off frekvens ved 50 Hz
4. (M) Programmet skal kunne vise blodtrykssignalet kontinuert
5. (M) Programmet skal programmeres i C#
6. (M) Programmet skal kunne lagre de målte data i en database
7. (S) Programmet bør kunne måle puls

Usability

1. (M) Blodtrykstallene der udskrives på brugergrænsefladen er røde
2. (S) Pulsmålingen burde kunne udskrives på brugergrænsefladen med grønne tal
3. (M) Brugergrænsefladen skal leve op til figuren udarbejdet i desgin afsnittet, se afsnit 3.3.1

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

Systemet får herved en tilgængelighed beregnet ved

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

Performance

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

Supportability

1. (M) Softwaren er opbygget af trelagsmodellen

Design 3

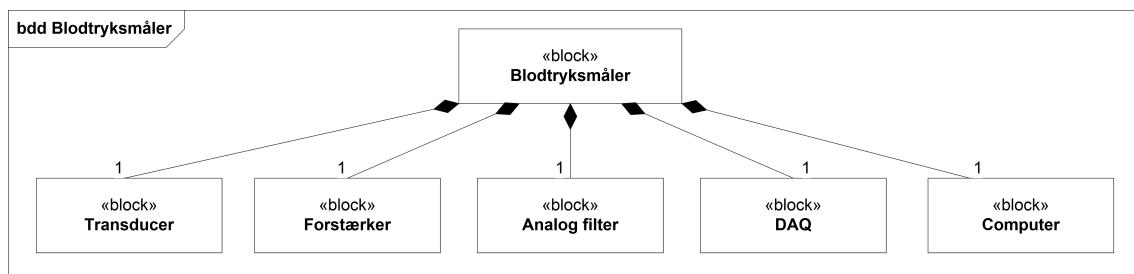
Version	Dato	Ansvarlig	Beskrivelse
---------	------	-----------	-------------

3.1 Indledning

Formålet med design afsnittet er at beskrive hardwaren og softwaren for blodtryksmåler-systemet. Systemet beskrives ved hjælp af udregninger, diagrammer, figurer og skitser som tydeliggører, hvordan de forskellige delelementers funktionalitet er, samt hvilke tanker der ligger til grund for den endelige implementering (se afsnit 4.1 for hardware og afsnit 4.2 for software).

3.2 Hardware arkitektur

I følgende afsnit beskrives hvordan blodtryksmåler systemet og dets delkomponenters opbygning.



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.



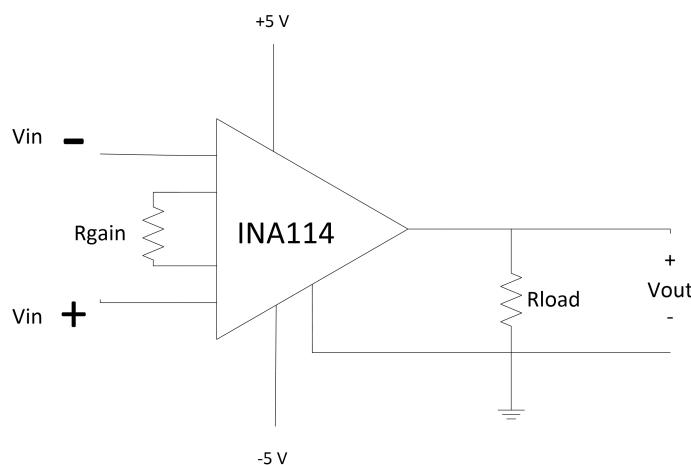
Figur 3.2: Internt blok diagram for blodtryksmåler systemet.

Ud af det interne blokdiagram, figur 3.2, kan ses det at blodtrykket i form af det målte tryk kommer ind i transduceren. Transduceren som omformer det målte tryk til et spændingssignal, sender signalet vidre 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 ikke er for lav. En Instrumentationsforstærkers indgangsimpedans i den virkelige verden, er højere, og den kan dermed opfange meget små signaler, som f.eks. blodtryk, der opererer i millivolt..

Vejleder rådede herefter til at projektgruppen brugte instrumentationsforstærkeren INA114. Forstærkerens design er valgt ud fra instrumenteringsforstærkerens datasheet's forslag, og kan ses på figur 3.3.



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 brøkregning, 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:

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

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 af typen Sallen-Key med unity gain, med båndbredde på 50 Hz (se 3.4). Desuden skulle filteret yderligere designes som et Butterworth filter med cut off frekvens på 50 Hz. C2 skulle vælges til at være 680 nF og gruppen fik desuden at vide, at R1 skulle være lig med R2. Operationsforstærkeren blev opgivet til at være af typen OP27.



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

Et Sallen-Keyfilter har en dæmpningsfaktor på 0,7, fordi at der, i et Sallen-Key filter, er blevet prioriteret et fladt frkvensområde, frem for hurtig dæmpning. Gruppen brugte en hjemmeside¹ som hjælpemiddel til at finde overføringsfunktionen for Sallen-Key lavpasfilteret. Denne ligning kan ses i ligning 3.5:

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

Da det er blevet opgivet at $R_1 = R_2$, kan overføringsfunktionen forkortes, som set på ligning 3.6.

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

Dernæst sammenlignes med standardformlen for overføringsfunktionen for et andet ordens filter på figur 3.7;

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

Ud fra dette kan regnes komponentværdierne for R, idet vi har en opgivet værdi for $\frac{2}{RC_2}$, som vist på figur 3.8

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

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

Gruppen brugte herefter MathCad til at isolere R, og herefter udregne værdien af denne. Dette kan ses på figur 3.9

$$\frac{2}{R \cdot 680 \cdot 10^{-9}} = 2 \cdot 0.7 \cdot (50 \cdot 2 \cdot \pi) \text{ solve}, R \rightarrow \frac{250000}{17 \cdot \pi} = 6.687 \times 10^3 \quad (3.9)$$

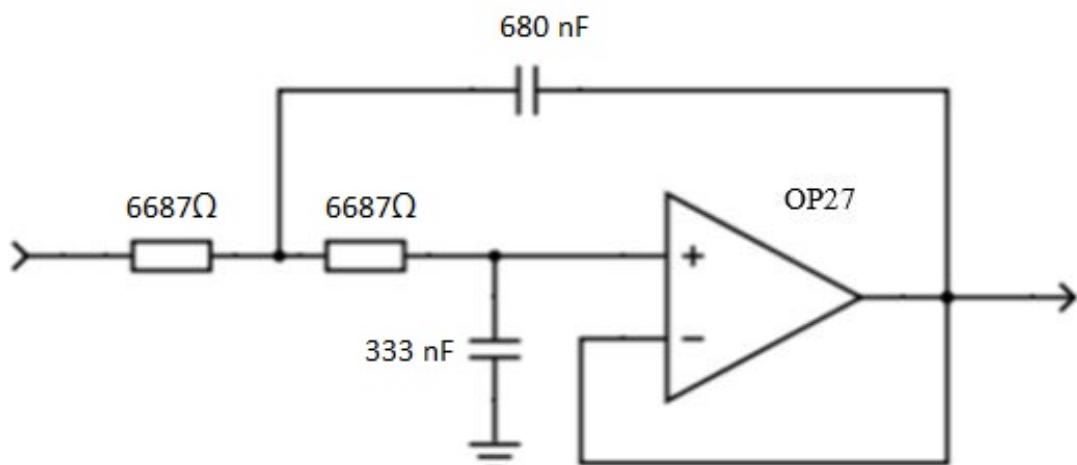
Dernæst kan komponentværdien for C1 udregnes, da gruppen nu havde en formel, men en ubekendt.

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

Ved hjælp af MathCAD isoleres C1.

$$\frac{1}{(6.687 \times 10^3)^2 \cdot (680 \cdot 10^{-9}) \cdot C_1} = (50 \cdot 2\pi)^2 \text{ solve}, R \rightarrow 333,2 \times 10^{-9} \quad (3.10)$$

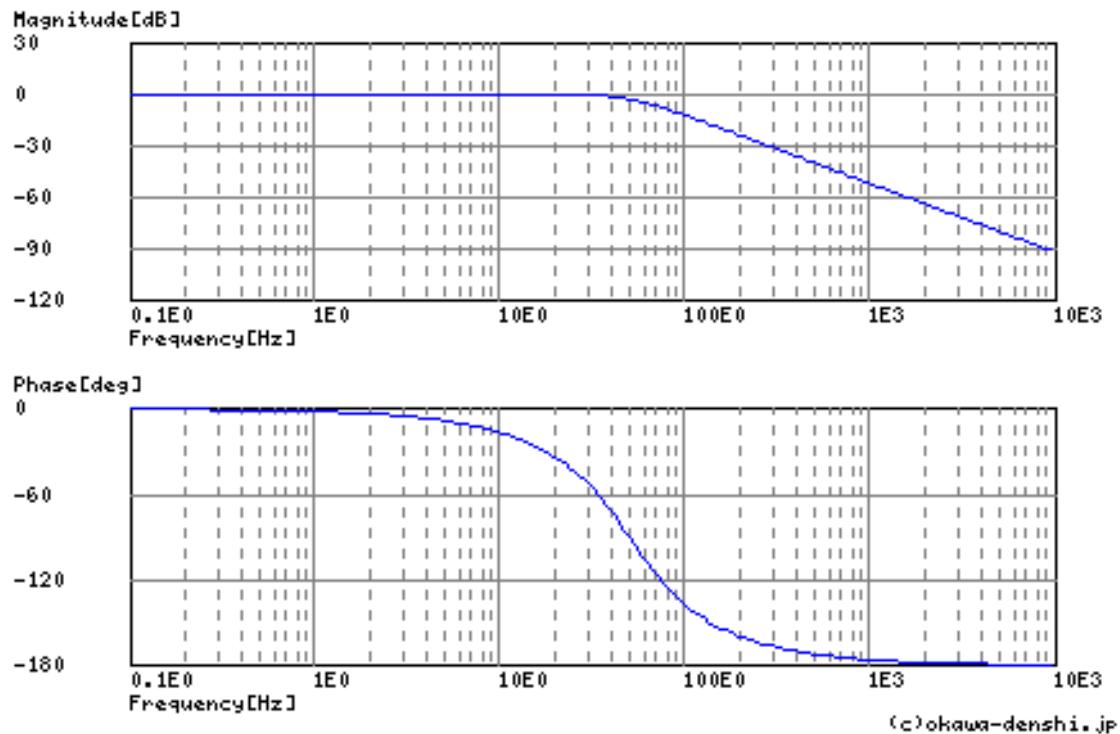
Derved er komponentværdierne for kredsløbet fundet og de ses indskrevet på 3.4.



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

For at underbygge gruppens teori omkring filteret, blev der ved hjælp af toollet "Sallen-Key Low-pass Filter Design Tool"² udarbejdet et bodeplot. Dette kan ses nedefor på figur 3.6.

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

Figur 3.6: Bodeplot af overføringsfunktionen³

3.2.3 Grænseflader

Navn	Input	Output	Interval	Beskrivelse
Transducer	Tryk	Spænding	Ind; -50 mmHg Ud; 0 til 6,25 mV	Transduceren, i form af en straingauge, reagerer i forhold til trykændringer, og udsender en spænding, som ændrer sig i forhold til tryk.
Forstærker	Spænding	Spænding	Ind; 0 til 6,25 mV Ud; -2,5V til 2,5V	Forstærkeren modtager det svage signal fra transduceren, og forstærker signalet op, så det matcher DAQ'ens dynamikområde.
Filter	Spænding	Analogt Signal	Ind; -2,5V til 2,5V Ud; -2,5V til 2,5V	Filteret modtager det forstærkede signal fra forstærkeren, og filtrerer signalet.
DAQ	Analogt Signal	Digitalt Signal	Ind; -2,5V til 2,5V Ud; -2,5V til 2,5V	DAQ'en konverterer det analoge signal fra filteret, om til et digitalt signal, som computeren modtager
Computer	Digitalt Signal	Grafisk Billede		Computeren modtager et digitalt signal fra DAQ'en, som bliver behandlet i koden.

Tabel 3.2: Grænsefladetabel

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

3.3 Software arkitektur

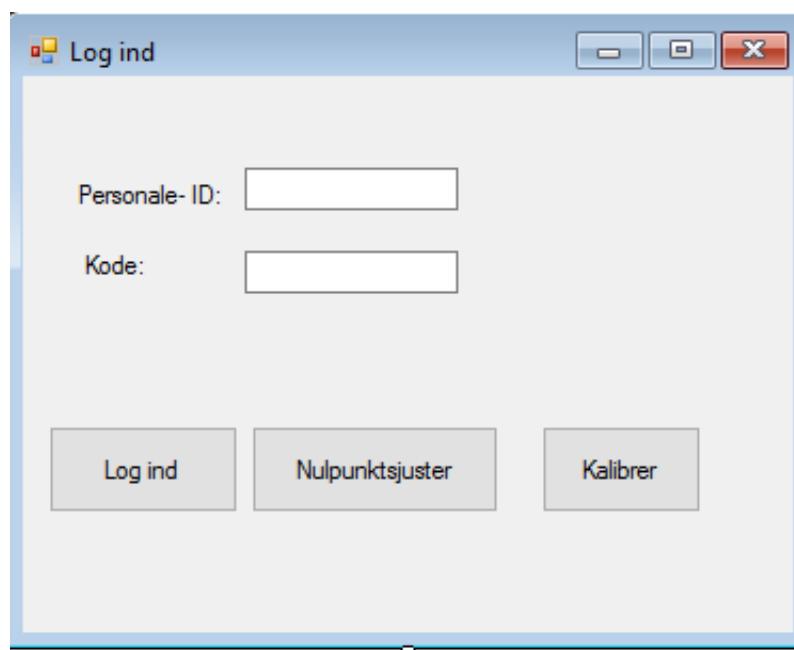
3.3.1 GUI

Dette afsnit beskriver hvilke tanker og overvejelser der er gjort i forbindelse med designet af diverse brugergrænseflader. Nedenfor ses udkast til disse brugergrænseflader. Den endelige udgave af brugergrænseflader kan ses i afsnit 4.2.

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.



Figur 3.7: Login vindue

I figur 3.7 er der tage 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

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

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.



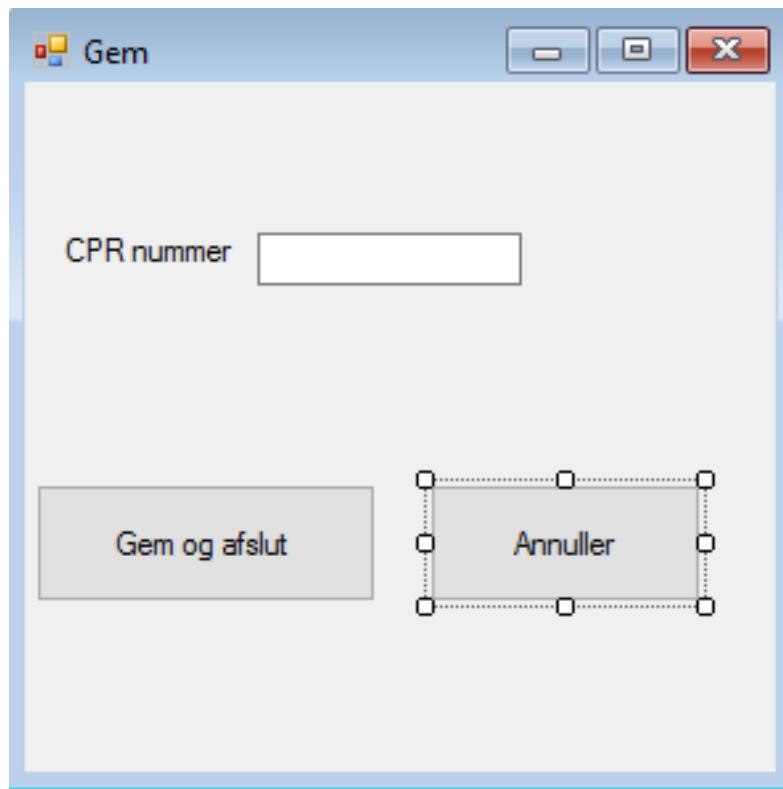
Figur 3.8: Kalibrer vindue

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 kontrollerer 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.

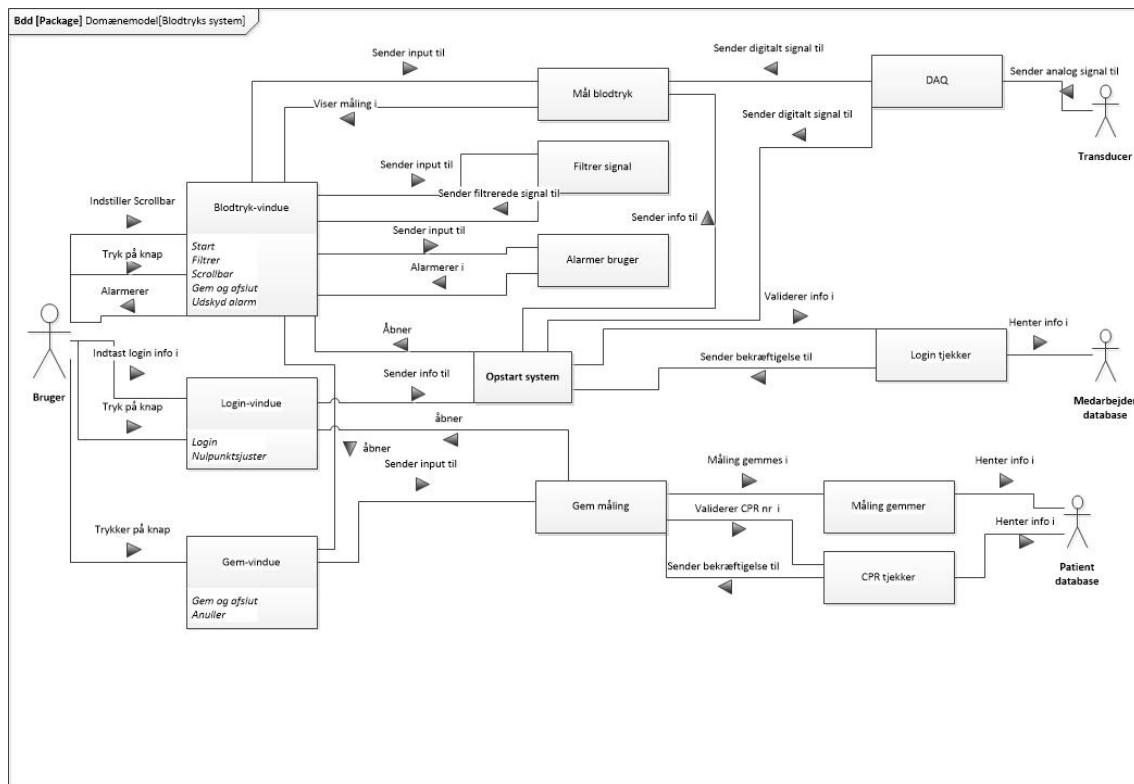


Figur 3.10: Gem vindue

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 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

3.3.3 Applikationsmodel

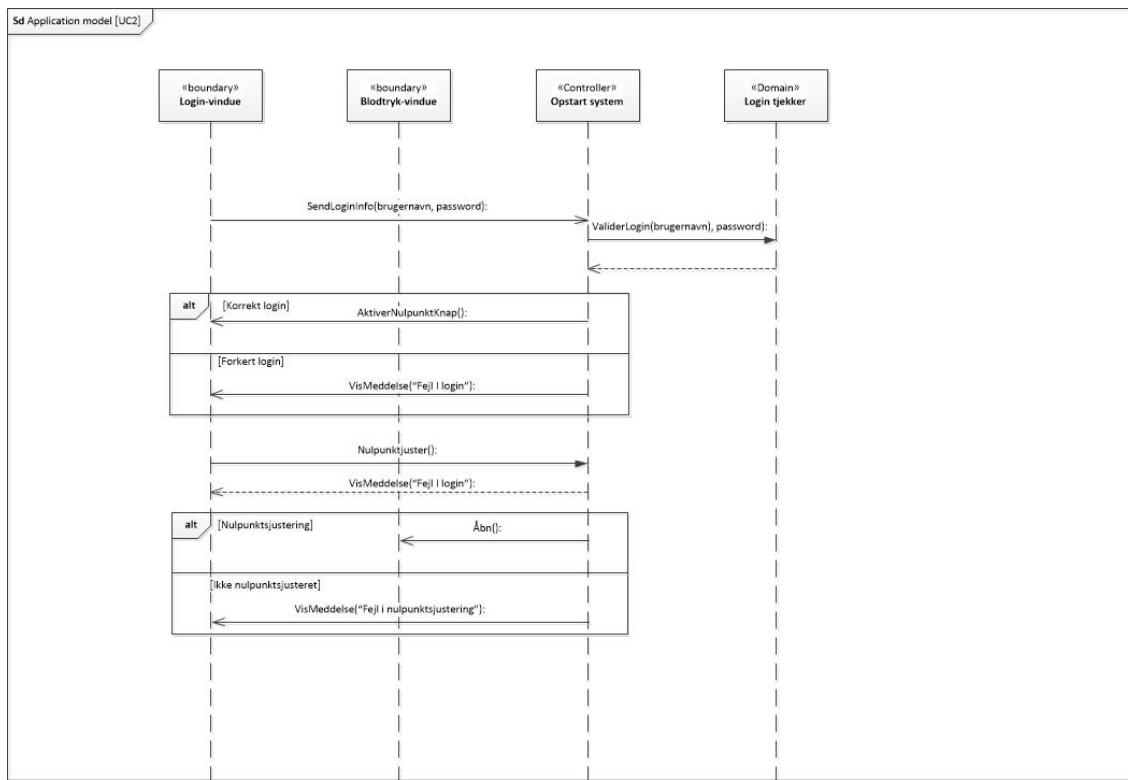
Applikationsmodellen er en model der på baggrund af domænemodel og Use cases udarbejdes software relaterede klasse applikationsmodeller, sekvensdiagrammer og opdaterede klasse applikationsmodeller.

Her er det ikke-opdateret klassediagram udeladt, hvorfor sekvensdiagram og opdaterede klasse applikationsmodel er udarbejdet på baggrund af fiktive metoder. De faktiske metoder kan ses i afsnit 4.2.5.

Sekvensdiagram

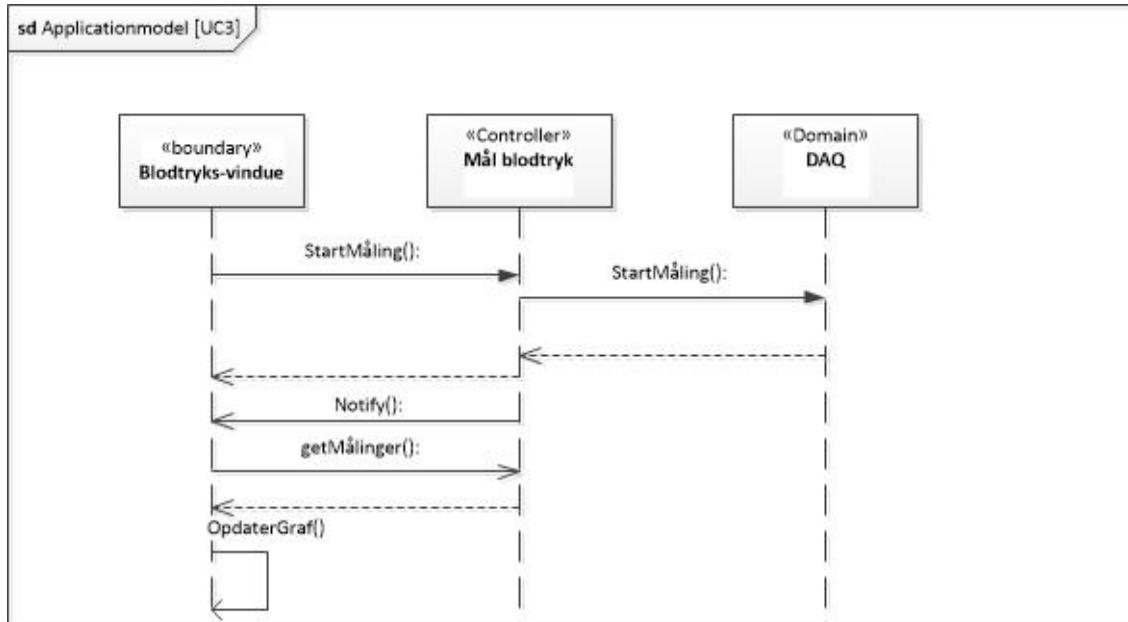
Sekvensdiagrammet er et interaktions diagram, der viser hvordan processer i systemet forløber. Use casene ligger til baggrund for udarbejdelsen af sekvensdiagrammerne.

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 metoder 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

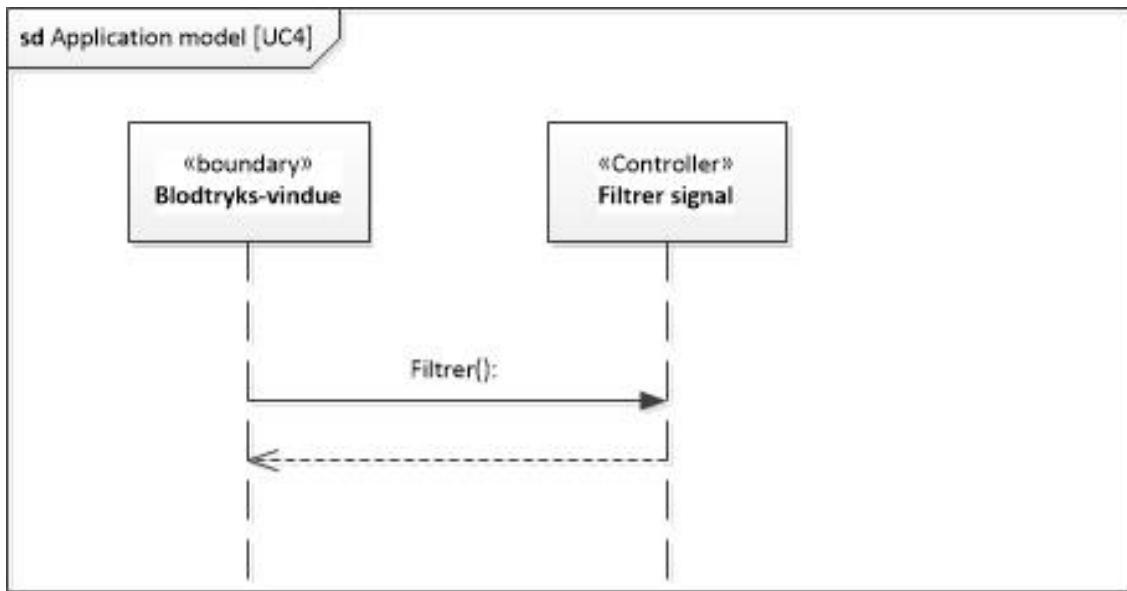
På figur 3.12 ses det hvordan brugeren logger ind i systemet og hvordan brugeren for systemet nulpunktsjusterer. Det er brugeren, der interagerer med systemet og er initiatør for at handlingerne bliver udført.



Figur 3.13: Sekvensdiagram UC3

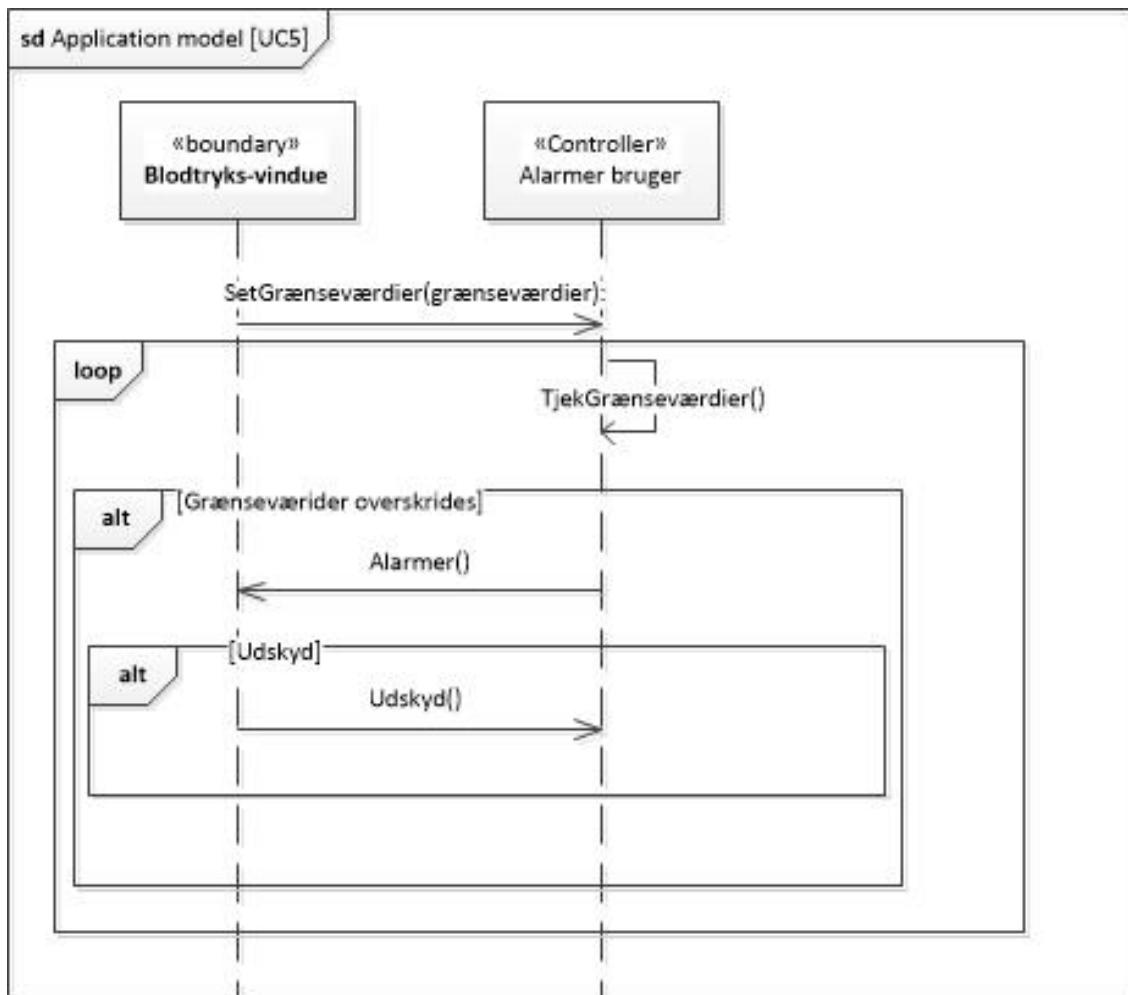
Figur 3.13 viser hvordan brugeren starter målingen og hvordan graferne vises på

brugergrænsefalden.



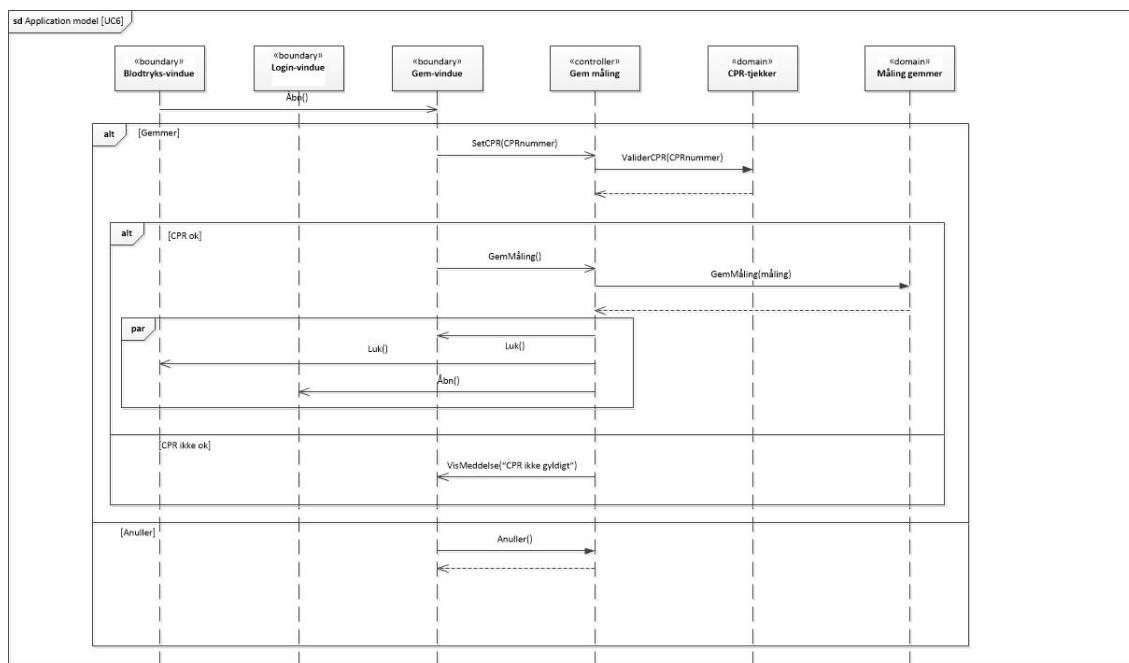
Figur 3.14: Sekvensdiagram UC4

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.

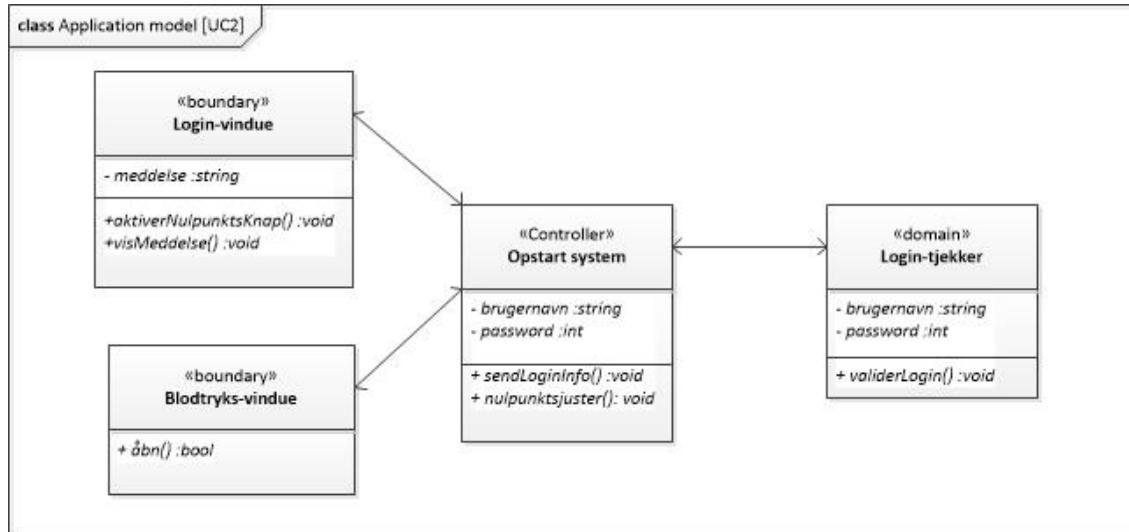


Figur 3.16: Sekvensdiagram UC6

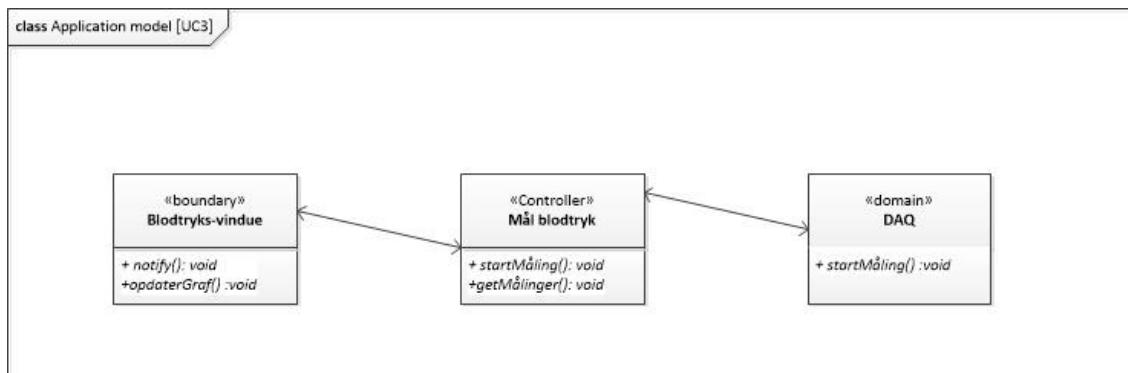
Figur 3.16 viser hvordan systemet gemmer og afslutter en måling.

Opdateret applikationsmodel

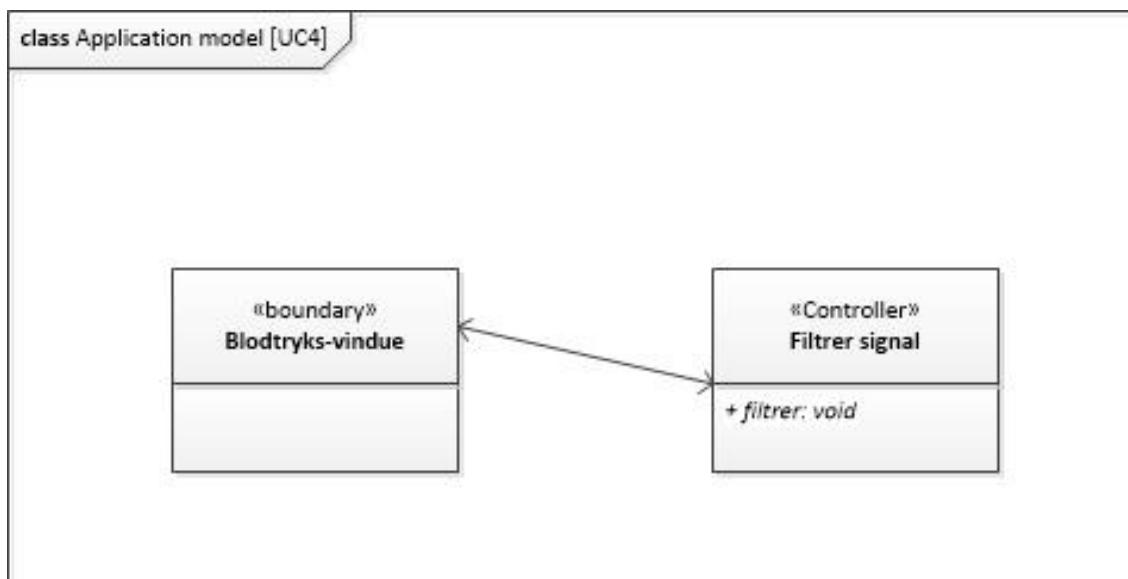
Klasse applikationsmodellerne er udarbejdet ud fra use casene, sekvensdiagrammer og domænemodellen. Der er ligesom ved sekvensdiagrammerne lavet et klassediagram for hver use case. Der er her også brugt virtuelle metoder.



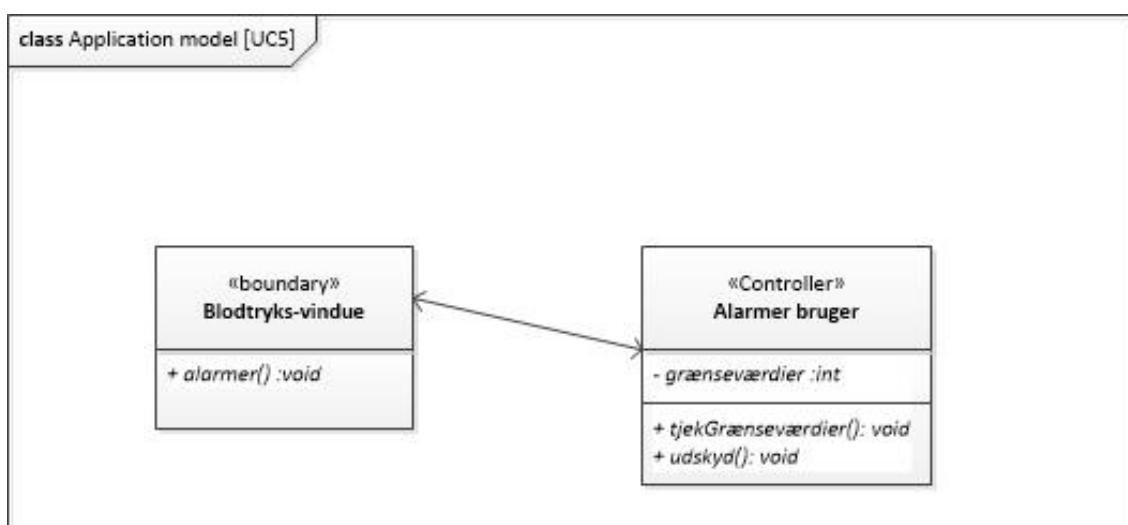
Figur 3.17: Klassediagram UC 2



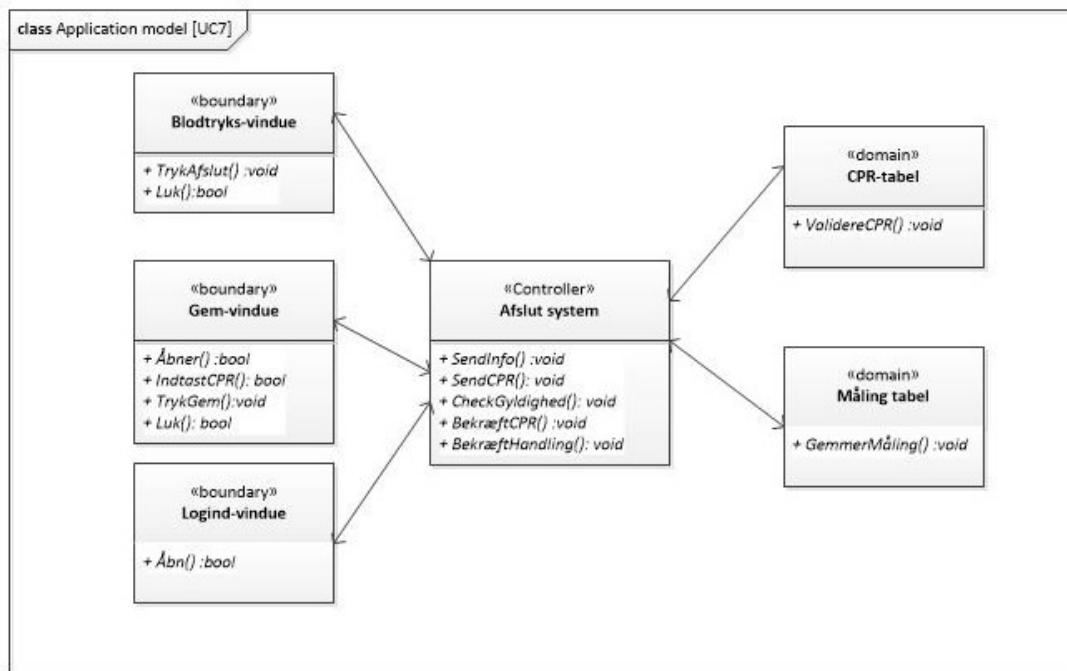
Figur 3.18: Klassediagram UC 3



Figur 3.19: Klassediagram UC 4



Figur 3.20: Klassediagram UC 5

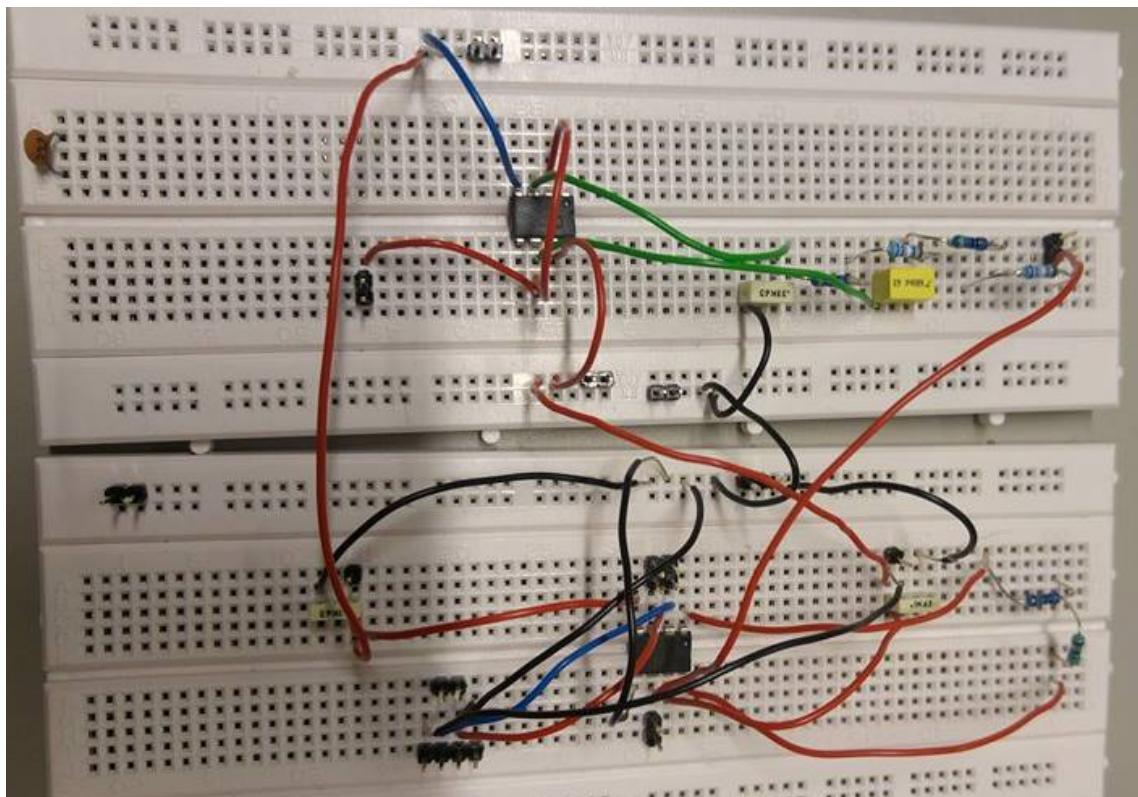


Figur 3.21: Klassediagram UC 7

Implementering 4

4.1 Hardware implementering

Efter teorifasen var ovre, blev de to blokke bygget op. Gruppen valgte at bygge forstærkeren og filteret på hver sit fumlebræt, dels på grund af pladsmangel og dels på grund af større sammenhæng mellem arkitekturen, og det endelige produkt.



Figur 4.1: Opstilling af forstærker og filter

Grundet mangel på præcise modstande, bedømte gruppen at det var bedst at bygge modstandende i forholdsvis filteret og forstærkeren, op i to, så gruppen kunne komme så tæt på den ønskede modstandsværdi som muligt.

Hardware overvejelser

I udviklingen af hardwaren, stødte gruppen på problematikker, som blev gjorde, at der var nogen komponenter, der blev nødt til at blive taget op til genovervejelse.

Et af de største problemer gruppen stødte ind i, i forhold til hardware, var brugen af en operationsforstærker i forstærkerblokken. En reel operationsforstærker har ikke en uendelig indgangs modstand, som i den ideelle verden, og der er derfor en risiko for at den lave spænding fra transduceren simpelthen ikke ville kunne passere forstærkeren. Gruppen valgte derfor at bruge en instrumenteringsforstærker i stedet, da den reelle komponents egenskaber ligger langt tættere på dens ideelle modpart, og derfor var bedre egnet i gruppens kredsløb. En anden fordel ved instrumenteringsforstærkeren er, at den er meget let at justere gain på.

En anden stor fordel ved instrumenteringsforstærkeren er, at common mode rejection er meget god. Common mode noise er støj, fra det omkringliggende miljø. Da en instrumenteringsforstærkers forstærkning kommer fra forskellen mellem dens to indgange, så kan støj påvirke forstærkningen, især når der arbejdes med så lave forstærkninger, som dem, der kommer fra transduceren. Ofte er der dog støj på begge indgange, og i den ideelle verden ville dette ikke påvirke forstærkningen. I den reelle verden kan støjen dog også ende med at blive forstærket. Common mode rejection beskriver hvor god en komponent er, til at sortere dette støj fra. Den brugte instrumenteringsforstærker har en common mode rejection på 120 dB, hvilket vurderes til en rimelig værdi.

En anden ting ved forstærkeren, som blev nødt til at blive ændret, var det dynamik område, som skulle udnyttes. Oprindeligt havde gruppen bestemt, at det filtrerede signal skulle ligge imellem -5V til +5V. Grundet en forstærkers manglende evne til at forstærke et signal op til dens forsyningsspænding, så blev gruppen nødt til at nedsætte signalområdet til -2,5V til +2,5V. Det var muligt at forstærke signalet yderligere, men grænsen blev sat til denne værdi, fordi signalområdet matcher et dynamikområde på AD konverteren.

Designet af filteret, lå fast på forhånd, og der var derfor ikke meget der kunne ændres i dette. Operationsforstærkeren var oplyst til at skulle være af typen OP27G, da denne ligeført har en god common mode rejection. Grundet mangel på præcise komponenter, har gruppen sat to modstande i serie, for at komme så tæt på den udregnede værdi, som muligt.

Gruppen valgte desuden at systemet skulle forsynes af Analog Discovery, frem for et batteri. Dette skyldes at Analog Discovery giver en stabil strøm, som ikke har brug for at blive afbalanceret af en spændingsudligner. Gruppen vurderede at usikkerhederne ville udligne hinanden, og så flere fordele ved at bruge Analog Discovery.

Implementering af forstærkeren

Den samlede stykliste for forstærkeren blev vist som på tabel 4.1.

Komponent	Antal	Type
Modstand	1	120 Ω
Modstand	1	4,8 Ω
Kondensator	2	100 nF
Instrumentationsforstærker	1	INA114

Tabel 4.1: Forstærkertabel

For forstærkeren gælder det at den beregnede R_{gain} er $125,31\Omega$ som det kan ses ud af komponentlisten består R_{gain} i praksis af to modstande på henholdsvis $4,8\Omega$ og 120Ω som er sat i serie. R_{gain} modstanden er i praksis $124,8\Omega$. I praksis er R_{gain} $0,51\Omega$ mindre end den i teorien skulle have været.

Implementering af filteret

Den samlede stykliste for filteret blev som vist på tabel 4.2.

Komponent	Antal	Type
Modstand	2	6,2 k Ω
Modstand	2	470 Ω
Kondensator	1	680 nF
Kondensator	1	330 nF
Operationsforstærker	1	OP27G

Tabel 4.2: Filtertabel

Det analoge filter består blandt andet af en 330 nF kondensator, C_1 , som i teorien er beregnet til at skulle have været $333,2$ nF. I praksis er kondensatoren C_1 $3,2$ nF mindre end den i teorien er beregnet til at skulle have været. Filteret består desuden af to modstande R_1 og R_2 som er identiske. I det realiserede analoge filter består hver modstand af to modstande på henholdsvis 6200Ω og 470Ω som er sat i serie. Dermed er både R_1 og R_2 6670Ω i praksis. I teorien var R_1 og R_2 udregnet til at skulle være 6687Ω . I praksis er der derfor 17Ω mindre end teorien foreskriver.

Generelt er der valgt at se bort fra de afvigelser, der er for komponentværdierne i praksis sammenlignet med de i teorien beregnet. Det er valgt da afvigelserne er relativt små i forhold til det pågældende komponent. For modstandende er der desuden 1 procents usikkerhed, hvilket betyder man alligevel ikke kan være helt sikker på komponentværdien.

På baggrund af de i praksis anvendte komponenter er den reelle knækfrekvens for det analoge filter beregnet. Til det formål er formlen som set på figur ?? anvendt.

$$f_c = \frac{1}{2\pi\sqrt{R_1C_1R_2C_2}} = \frac{1}{2\pi\sqrt{6687 \cdot 333,2 \times 10^{-9} \cdot 6687 \cdot 680 \times 10^{-9}}} = 50,37 Hz$$
(4.1)

Desuden er den reelle ζ for dette filter, ifølge Okawadenshi, 0,697.

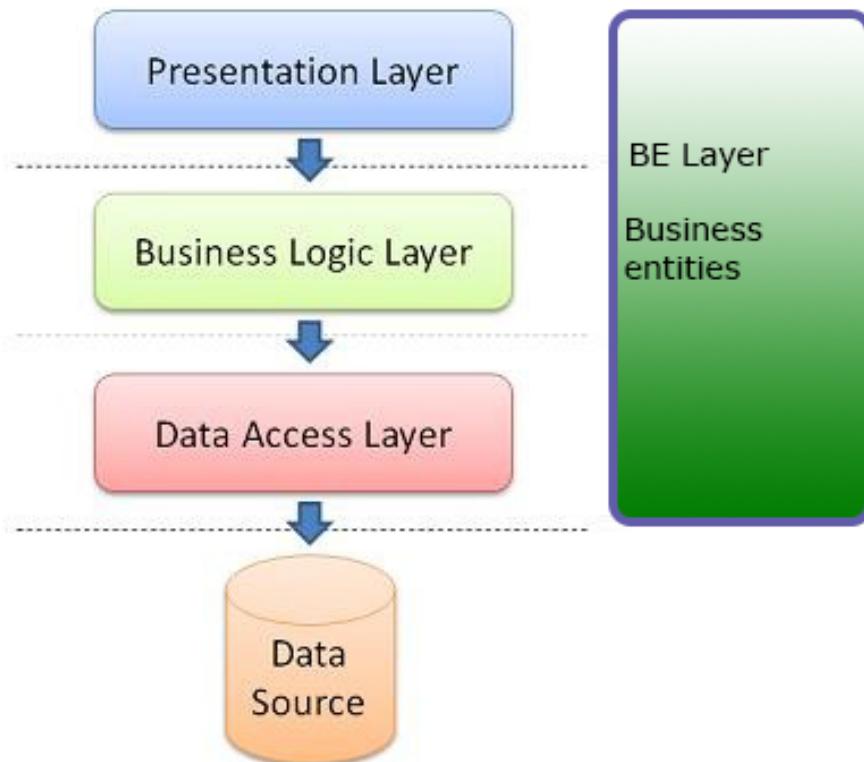
4.2 Software implementering

Implementerings afsnittet beskriver hvilke faktiske handlinger der er foretaget ud fra arkitektur og design afsnittet. Yderligere beskriver afsnittet hvilke software orienterede metoder der er benyttet for at løse problemstillingen.

4.2.1 3-lagsmodellen

Softwareen er implementeret først og fremmest ud fra 3-lagsmodellen 3-lagsmodellen er bestående af lagene præsentationslag, logiklag og datalag. Denne opdeling af lagene gør det langt lettere at vedligeholde systemet fordi der kan ændres i et enkelt lag uden det har indflydelse på resten af programmet.

Det er desuden en god software arkitektur at bruge ved et system udarbejdet af en gruppe, da der kan arbejdes på to forskellige lag af to personer samtidigt, hvis bare grænsefladerne bliver overholdt. 3-lagsmodellen er bygget op som vist på figur 4.2:

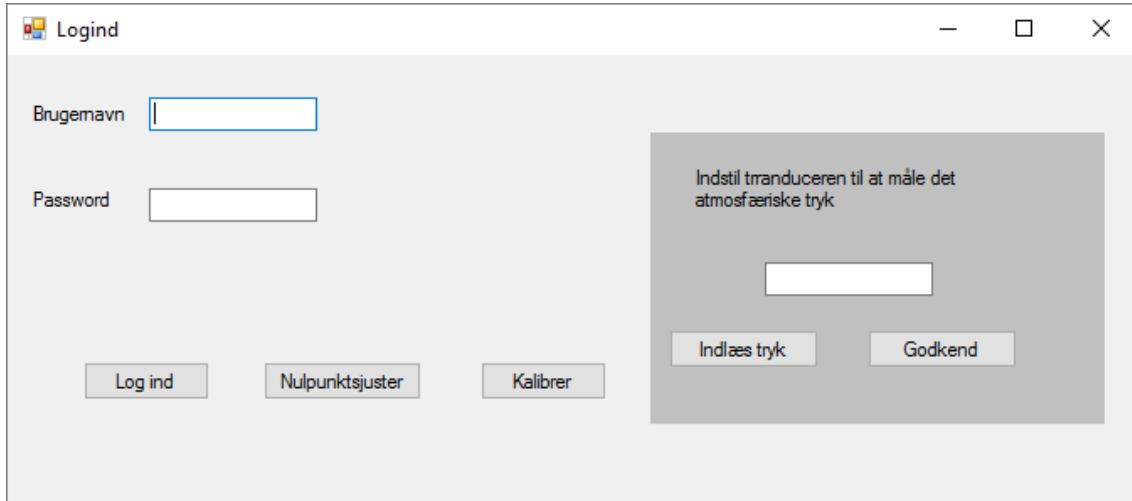


Figur 4.2: 3-lagsmodellen

Præsentationslag

Som vist på figur 4.2, er blodtrykssystemet bygget op med først et præsentationslag bestående af tre brugergrænseflader, som brugeren kan interagere med.

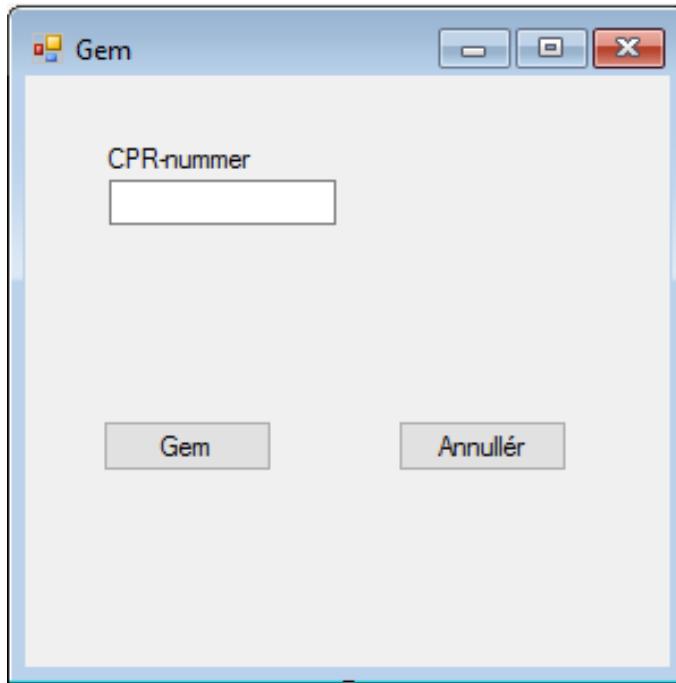
Login brugergrænseflade, bestående af tekstbokse til brugernavn og kodeord, samt paneler til både kalibrering og nulpunktsjusteringen, som vist på figur 4.3:



Figur 4.3: Login GUI

Dernæst åbnes hoved brugergrænsefladen, hvor blodtrykssignalet vises grafisk, samt brugeren har mulighed for at benytte en række funktioner i form af knapper og trackbars, som set på figur ??

Til sidst vises en brugergrænseflade for gemmefunktionen, hvor brugeren skal indtaste et CPR-nummer, samt bekræfte afslutning af målingen, se figur 4.4:



Figur 4.4: Gem GUI

Logiklag

Logik laget er det lag der spiller sammen med både data- og præsentationslag. Sammenspillet med præsentationslaget fungerer ved alt brugerens interaktion med systemet bearbejdes i logiklaget. Et eksempel herpå er det digitale filter, hvor brugeren gennem brugergrænsefladen tilslutter filteret. Dette igangsætter en metode i logik laget, som sørger for at signalet filtreres. Metoden er vist i figur 4.6 og figur 4.5:

```
if (filterOn)
{
    List<double> filterliste = new List<double>();
    for (int i = (grafcount < 5 ? 0 : grafcount - 5); i < grafcount; i++)
    {
        filterliste.Add(ufiltreredeGrafværdier[i]);
    }
    grafværdier[grafcount] = filter.filtrer(filterliste, ufiltreredeGrafværdier[0]);
}
```

Figur 4.5: Filtrer i filter klassen

```

1 reference
public double filtrer(List<double> liste, double lastValue)
{
    if (liste.Count > 0)
    {
        return liste.Average();
    }

    return lastValue;
}

```

Figur 4.6: Filtrer i filter klassen

Figur 4.5 viser hvordan der i logikklassen oprettes en liste med **JEPPE!!!!**

Herefter bliver der i figur 4.6 taget gennemsnit af listen hvorefter den sidste værdi returneres, og derefter udskrives. Filteret er deraf et FIR filter af typen moving average.

Datalag

Datalaget er det lag som spiller sammen med databasen og logiklaget. Dvs. at datalaget modtager bruger inputs fra brugergrænsefladen gennem logiklaget. Herefter sender og validerer datalaget informationerne i en database. Eksempelvis er bliver login oplysninger tjekket i databasen via datalaget, dette er vist i figur 4.7:

```

1 reference
public bool checkLogin(string username, string password)
{
    bool resultat = false;
    cmd = new SqlCommand("select kodeord from personale where personale_id ='" + username);
    conn.Open();

    // Udfør det ønskede SQL statement på DB
    rdr = cmd.ExecuteReader();
    if (rdr.Read())
    {
        resultat = (password.Equals(rdr.GetString(0)));
    }

    conn.Close();
    return resultat;
}

```

Figur 4.7: Login tjekker metode

Figur 4.7 viser hvordan login tjekker metoden opretter forbindelse til en lokal database, hvorefter de indtastede værdier tjekkes i databasen. Til slut lukkes forbindelsen igen. Værdierne der tjekkes efter med kan ses i personale tabellen i databasen, som vist i figur 4.8

	personale_id	kodeord
▶	1234	fido
*	NULL	NULL

Figur 4.8: Personale tabel

Herudover er det også i datalaget at data bliver indlæst fra DAQen. Dette gøres ved at udnytte, at DAQen har et asynkront kald i sig. Dette asynkrone kald gør, at der bliver leveret data fra DAQ til datalag løbende istedet for i intervaller, som ville være tilfældet ved synkrone kald. Det asynkrone kald tillader altså at portioner af koden bliver udført løbende. Hvis ikke det asynkrone kald blev benyttet her ville der gå data tabt.

4.2.2 Tråde

Tråd programmering benyttes, når man som her ønsker flere opgaver i programmet udført løbende. Dette kan gøre sig gældende eksempelvis ved at programmet ønsker at indlæse data og vise det i en graf, samtidig med at brugeren har adgang til alle interface funktioner. Det er disse delprocesser, som udgøre trådene. Hvis en enkelt tråd skulle gå i stå, skal de andre tråde stadig køre, således at programmet stadig kører videre.

4.2.3 Observer

4.2.4 Push pull

4.2.5 UML klassediagram

4.2.6 Kode udsnit

4.2.7 Diagrammer

Test 5

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

5.1 Indledning

Test afsnittet giver en beskrivelse af hvilke test der løbende er blevet foretaget på både hardware og software. Testene består af modul test, hvor enkelte delelementer af systemet uafhængigt af hinanden. Modul testen er til for at verificere de enkelte enheder virker efter hensigten.

En mere omfattende integrationstest, hvor de større dele af både hardware og software testes, afhængigt af hinanden. Ved integrationstesten ligger fokus på hvordan systemet fungerer, set fra udviklers synspunkt.

Begge test ligger til grund for den endelige accepttest, hvor testen ses fra kundens synspunkt.

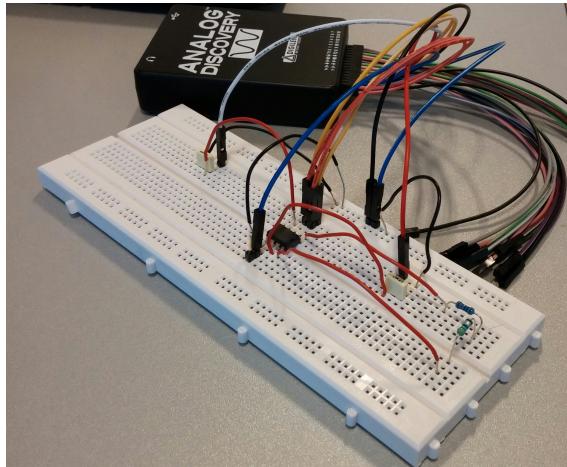
5.2 Modul test hardware

Til test af hardwaredelen blev der først udført modultest på henholdsvis forstærkeren og det analoge filter. Efterfølgende blev der udført integrationstest på forstærkeren og det analoge filter sat sammen med tryk transduceren. Slutteligt blev den samlede hardwaredel bestående af tryktransducer, forstærker og analogt filter testet på en vandsøje.

5.2.1 Modul test af forstærkeren

I modultesten af forstærkeren blev Analog Discovery brugt som spændingsforsyning og oscilloskop på to forskellige målepunkter placeret ved henholdsvis indgangs signalet og udgangssignalet.

Da tryktransduceren forventes at have output spændinger i området 0 til 6,25 mV blev signalet fra transduceren simuleret ved en sinus på 1 Hz med et offset på 0. Den laveste amplitude for indgangs signalet var 1 mV. For hver måling blev amplituden for indgangs signalet sat op med 1 mV indtil den sidste måling hvor amplituden var nået op på 10 mV. Således blev der 10 forskellige målinger. Reelt set var det kun nødvendigt at teste op til 6 mV eller 7 mV, da det er herimellem, at man kan forvente et max tryk fra transduceren at ligge. Da der alligevel er taget målepunkter op til 10 mV skyldes det at der ved flere målinger kan laves en ”pænere” tendenslinje ved lineær regression.



Figur 5.1: Måleopstilling ved modul test af forstærkeren.

Ud af målingerne blev der foretaget lineærregression over de 10 målepunkter. Tendenslinjen der kom ud af den lineæreregression blev som følger:

$$Y = 415,8 \cdot x - 0,09 \text{ hvor } R^2 = 0,99 \quad (5.1)$$

Her kan Y beskrives som værende forstærkningen ved en given frekvens. Konstanten 415,8 er den reelle forstærkning som er målt for forstærkeren. Skæringen med y-aksen burde være 0, men grundet måleusikkerheder er den blevet -0,09, hvilket også er acceptabelt. Den høje R^2 værdi indikerer, at der er en tydelig lineær sammenhænge mellem den påtrykte spænding og spændingen af output, dvs. forstærkningen er lineær.

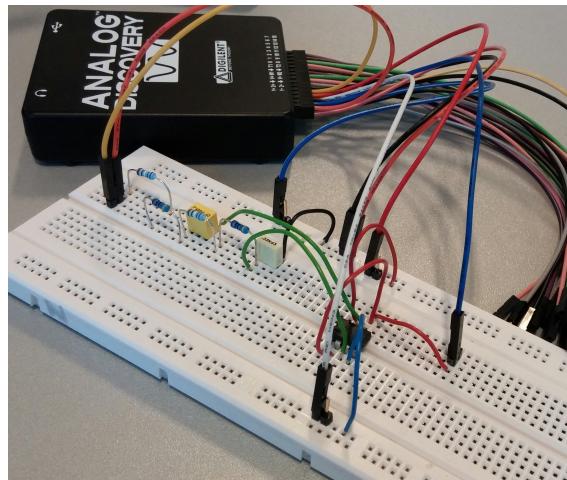
Senere påtryktes samme målopstilling et DC-signal med amplituder startende fra 3mV op til 10 mV. Igen blev amplituden øget med 1 mV for hvert måling således, at der blev 7 målepunkter. Ved målinger foretaget på signaler med både 1 mV og 2 mV er offsettet i Analog Discovery så stort, at målingerne er umulige. Efter at have foretaget lineærregression på de 7 målepunkter så tendenslinjen ud som følger:

$$Y = 386,38 \cdot x + 0,5547 \text{ hvor } R^2 = 0,998 \quad (5.2)$$

Forstærkningen givet ud fra de 7 måle punkter er på 386 hvilket er en del under den målte forstærkning når systemet påtrykkes et sinussignal i stedet for et DC-signal. Dette kan skyldes at der i målingen af DC-signalet er færre målepunkter, og at det dermed bliver en mere usikker måling, da selv en mindre måleusikkerhed har en større indflydelse på den målte forstærkning.

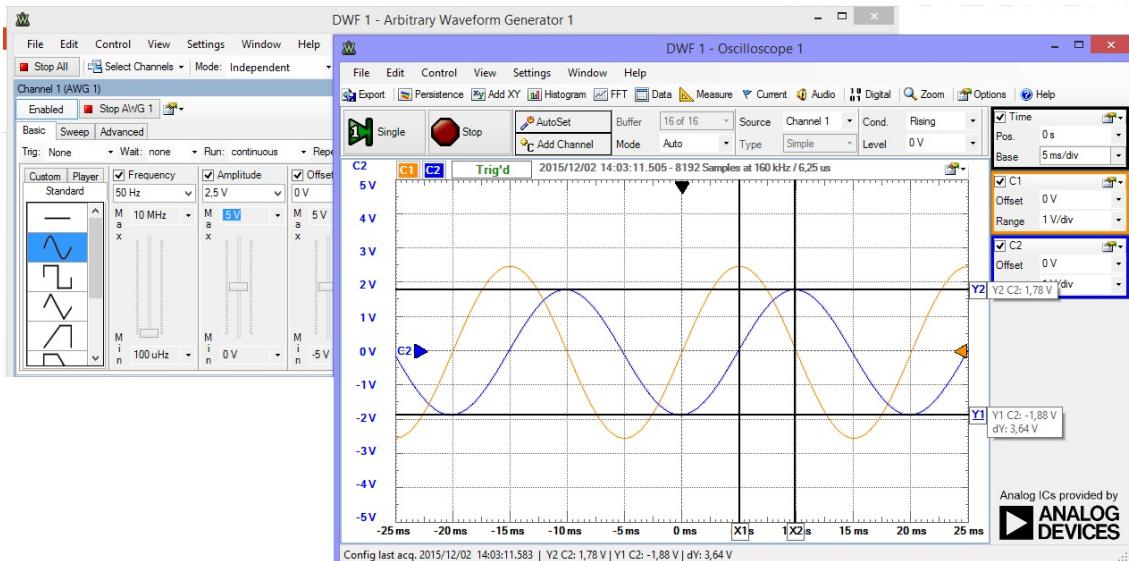
5.2.2 Modul test af det analoge filter

I modultesten af det analoge filter blev Analog Discovery brugt som spændingsforsyning, signal generator og som oscilloskop til at måle amplituderne for input signalet og output signalet for filteret.



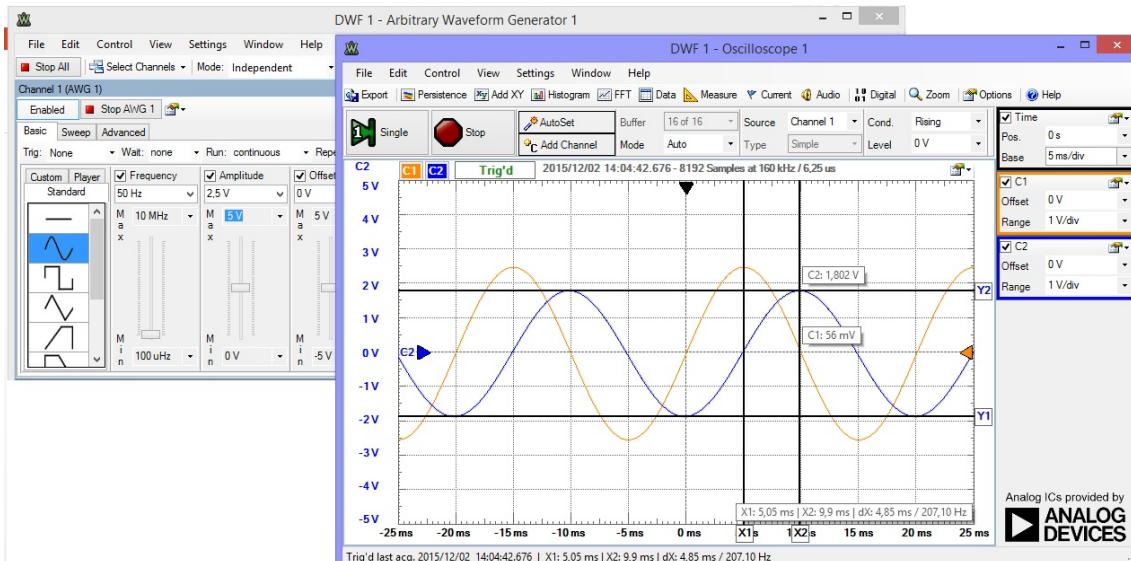
Figur 5.2: Måleopstilling ved modultest af det analoge filter.

I modultesten af det analoge filter ændres frekvensen for det påtrykte sinussignal på filter indgangen. Der blev målt på i alt 21 målepunkter som lå i intervallet 1 Hz til 500 Hz, se bilag XX for de præcise angivelser af de udvalgte målefrekvenser. Output amplituden samt Δt mellem de to grafer blev aflæst på de to grafer. På baggrund af disse målinger blev dæmpningen af signalet afbilledet.



Figur 5.3: Aflæsning af amplitude størrelse for outputsignalet, C2, fra filteret når inputsignalet, C1, har en amplitude på 2,5 V

Herefter blev tidsforskydningen aflæst således at fasedrejet kunne udregnes.



Figur 5.4: Aflæsning af tidsforskydningen for outputsignalet, C2, fra filteret set i forhold til indgangssignalet C1

Det gælder generelt for et 2. ordensfilter af standarttypen som der er arbejdet med igennem projektet, at fasen er 0° en dekade før filterets knækfrekvens og falder med $-90^\circ/\text{dekade}$ frem til dekaden efter knækfrekvensen. Derfor skal filteret i praksis have et fasedrej på minus -90° ved knækfrekvensen 50 Hz. Som det ses ud graf!! er fasedrejet ved frekvensen 5 Hz $-5,2^\circ$, denne skulle reelt set have været 0° . Ved knækfrekvensen som er sat til 50 Hz er fasedrejet $-86,4^\circ$, hvor fasedrejet skulle have været -90° . Ved målingen for 53 Hz er fasedrejet $-95,4^\circ$. Dermed kan der argumenteres for, at filterets reelle knækfrekvens må ligge et sted imellem 50 Hz og 53 Hz. For målingen på 500 Hz er fasedrejet -180° , hvilket stemmer overens med teorien for 2. ordensfilterets fasekarakteristik.

5.3 Modul test software

Til test af softwaren er der først og fremmest foretaget modul test af kodeelementer, for at teste om de fungerer efter hensigten. Dette er gjort ved hjælp af debug funktionen i Visual Studio. Disse test er foretaget løbende i udviklingsprocessen, hver gang en metode er tilføjet eller ændret. De væsenligste dele er nedenfor beskrevet:

Login funktionen er den første funktion, der testes ved sammenligne tilladte brugernavne og kodeord i databasen med indtastede. Dvs. at kun de brugernavne og kodeords par der findes i databasen, som giver adgang til systemet.

Nulpunktsjusteringen er testet ved at påsætte systemet en DC spænding på 2 Volt, hvorefter det indlæste tryk aflæses og sammenlignes med de 2 Volt.

I forbindelse med test af nulpunktjusteringen, hvor der påsættes 2 Volts DC spænding, er kalibreringen også testet ved at indtaste kalibreringsfaktoren på 2. Herefter er tjekkes at signalet er forstærket 2 gange.

Testene foretaget i forbindelse med indlæsning af blodtrykssignal og detektering af systole, diastole og puls er alle foretaget ved følgende:

Der indsættes et kendt blodtrykssignal, med kendt antal toppe og dermed kendt systole,

diastole og puls. De kendte værdier sammenlignes med værdierne der vises grafisk, samt med værdierne der findes ved debugging af systemet.

Alermen er tjekket ved at påsætte et blodtrykssignal med kendte blodtryksværdier: systolisk > 160 eller < 100 og diastolisk > 100 eller under 40), hvorefter det er testet om alermen lyder ved de fire tilfælde uafhængigt af hinanden.

Yderligere for alermen er der justering af grænseværdierne testet, ved at ændre disse grænseværdier således ovenstående værdier ikke er kritiske, for herefter at tjekke om alermen lyder.

Ved alarm, er det testet at lyden kan udskydes med 3 minutter, som skrevet i koden, ved at lade blodtrykssignalet overskride grænseværdierne i mere end 3 minutter.

Gemmefunktionen er testet ved at sende et kendt blodtrykssignal igennem systemet, og tjekke om de data der gemmes i databasen stemmer overen med de kendte data. I samme forbindelse er CPR tjekker testet ved først at indskrive et gyldigt CPR nummer, som først tjekkes i CPR-tjekker metoden og dernæst i databasen. Derefter er samme procedure gentaget med et ikke gyldigt CPR-nummer.

5.4 Integrationstest hardware

5.5 Integrationstest software

5.6 Integrationstest system

Accepttest 6

Version Dato Ansvarlig Beskrivelse

6.1 Indledning

I accepttesten testes de krav, der er opstillet i kravspecifikationen. Det fremgår af accepttesten 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 accepttest af use casene altså de funktionelle krav efterfulgt af accepttest af de ikke funktionelle krav.

Til test af use case 2-6 er der benyttet et signal fra PhysioNet. Signalet er et blodtrykssignal, som er vedlagt i bilag, se bilag XX.

6.2 Accepttest af Use Cases

6.2.1 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	Systemet er forbundet til en kendt kilde	✓
2. Mål det atmosfæriske tryk	Det atmosfæriske tryk kan aflæses	Det atmosfæriske tryk kan aflæses	✓
3. Aflæs spændingen målt på udgangen af det analoge filter	At der forekommer en værdi, som efterfølgende kan bruges til lineærregression		✓

4.	Skriv resultatet ned og gentag punkt 1 og 2. med to nye tryk	Der er tre værdier skrevet ned	Nye værdier kan notes	✓
5.	Foretag lineærregression	Afvigelseskoefficient er fundet	Afvigelseskoefficient noteres	✓
6.	Indtast afvigelses koeficient i kalibrering panel i "Log ind"-vinduet	Værdien er synlig i tekstboksen	Værdien er synlig i tekstboksen	✓
7.	Tryk på "Kalibrer"-knappen	Systemet er kalibreret	Systemet kalibreret	✓

Tabel 6.2: Accepttest af Use Case 1.

6.2.2 Use Case 2

Opstart system

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1.	Indstil transducer til at måle atmosfærisk tryk	Transduceren er instillet	Atmosfærsik tryk måles
2.	Tryk på "Nulpunktsjuster"-knap	Panel vises med instruktioner om nulpunktsjustering	Panel vises med instruktioner om nulpunktsjustering
3.	Tryk på "indlæs tryk"-knap	Et tryk vises i tekstdokumentet	Tryk vises
4.	Tryk på "Godkend"-knap	Panelet lukkes	Panelet lukkes

5.	Indtast personale-ID i brugenavnsfeltet; "1234" og indtast personlig kode i kodeordsfeltet; "fido"	Loginoplysninger bliver udfyldt	De rigtige oplysninger vises	✓
6.	Tryk på "Log ind"-knappen	Log ind oplysninger er gyldige og stemmer overens med hinanden. "Blodtryks"-vinduet vises	Login vinduet lukkes og blodtryk vinduet vises	✓

Undtagelser

6.a	1. Indtast personale-ID i brugenavnsfeltet; "fido" og indtast personlig kode i kodeordsfeltet; "1234"	Nye log ind oplysninger vises	Nye log ind oplysninger vises	✓
	2. Tryk "Log ind"	Besked om at de ikke findes vises	Besked vises	✓

Tabel 6.3: Accepttest af Use Case 2.

6.2.3 Use Case 3**Mål blodtryk**

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Påsæt signal fra Phy- sioNet	Signalet fungerer	Signalet fungerer	✓
2. Tryk på "start"-knappen i i blodtryks- vinduet	Graf og blodtryks værdier vises på bruger- grænsefladen	Graf og værdier vises	✓

Tabel 6.4: Accepttest af Use Case 3.

6.2.4 Use Case 4

Filtrer signal

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Påsæt realistisk signal fra fysionet	Signal vises i grafen	Signalet vises med støj	✓
2. Tryk på "Til"-knappen under filter	Signalet udglattes	Signalet udglattes	✓
3. Tryk på "Fra"-knappen under filter	Signal udglattes ikke	Signalet udglattes ikke	✓

Tabel 6.5: Accepttest af Use Case 4.

6.2.5 Use Case 5

Alarmer bruger

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Indstil cursor til 100 for diastolisk øvre grænse	Der står 100 i pågældende tekst felt	Der står 100 i pågældende tekst felt	✓
2. Indstil cursor til 80 for diastolisk nedre grænse	Der står 80 i pågældende tekst felt	Der står 80 i pågældende tekst felt	✓
3. Indstil cursor til 100 for systolisk øvre grænse	Der står 100 i pågældende tekst felt	Der står 100 i pågældende tekst felt	✓
4. Indstil cursor til 80 for systolisk nedre grænse	Der står 80 i pågældende tekst felt	Der står 80 i pågældende tekst felt	✓

5.	Tryk på 'Juster' knappen i blodtryksvindue	Blodtryksmåling startes og Alarm går igang	Blodtryksmåling startes og Alarm går igang	✓
<i>Undtagelser</i>				
5a	Tryk på "udskyd alarm"-knappen	Alarmen udskydes med 3 minutter	Alarmen udskydes med 3 minutter	✓

Tabel 6.6: Accepttest af Use Case 5.

6.2.6 Use Case 6

Afslut system

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1.	Tryk på "Afslut måling"-knappen	"Gemme"-vinduet vises	"Gemme"-vinduet vises
2.	Indtast CPR-nr "1111111118"	CPR-nummeret synligt i pågældende tekst felt	CPR-nummeret synligt i pågældende tekst felt
3.	Tryk på "Gem"-knappen	Bekræftigelse vises	Bekræftigelse vises
3.	Tryk på "Ok"-knappen	"Gemme"-vindue og "Blodtryks--vinduet lukkes. "Login"-vinduet vises	"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	"Gemme"-vinduet lukkes og "Blodtryk"-vinduet vises

2.a. Indtast nummeret "1234567890"	CPR-	Beskeden "CPR ikke gyldigt." vises	Beskeden "CPR ikke gyldigt." vises	✓
------------------------------------	------	------------------------------------	------------------------------------	---

Tabel 6.7: Accepttest af Use Case 7.

6.3 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 30 sekunder efter opstart	Start programmet, hvorefter der vha. stopur måles opstarts-tiden	At programmet er opstartet og ny måling er igang efter 30 sekunder	15 sekunder	✓
2. Systemet skal kunne forstærke signalet fra transduceren 400 gange	Start systemet	At signalet er forstærket	Egentlige test ligger i integrationstest, se afsnit 5.2	(✓)
2. Systemet skal kunne forstærke signalet med det indbyggede analoge antialiaserings filter med en båndbredde på 50 Hz	Start systemet	At signalet er forstærket	Egentlige test ligger i integrationstest, se afsnit 5.2	(✓)
3. Programmet skal kunne vise blodtrykssignalet kontinuert	Tryk på "Start måling"-knap	At blodtryks-signalet er vist kontinuert på brugergrænse-fladen	Blodtrykssignalet er vist kontinuert på brugergrænsefladen	✓

4. Programmet skal program- meres i C# Start program- met At koden er i C# Programmet er i C# ✓

5. Programmet bør kunne måle og afbillede puls Tryk på "Start måling" At pulsen er af- billedet på bru- gergrænseflade Pulsen af afbil- ledet ✓

Usability

1. Blodtrykstallene der udskrives på bruger- grænseflade er røde Tryk "Start måling"-knap At blodtrykstal- lene er røde Blodtrykstallene er røde ✓

2. Pulsmålingen skal udskrives på brugergræn- sefladen med grønne tal Tryk "Start måling"-knap At pulsen vises med grønne tal Pulsen vises med grønne tal ✓

3. Brugergrænseflade lever op til figu- ren udarbejdet i design afsnittet, se afsnit 3.3.1 Opstart pro- gram og log ind At brugergræn- seflade inde- holder samtlige funktioner som på figuren 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 Kan ikke testes

2. Systemet skal have en "mean time to resto- re" på højst 24 timer Start systemet og herefter gen- start, hvor der tages tid med et stopur At programmet er klar igen inden for 24 timer Kan ikke testes

Performance

1. Systemet bør kunne gemme data på 5 sekunder +/- 10% Tryk på "Gem og afslut"-knap og tag tid med sekunder +/-. stopur At data er inden for 5 sekunder Afhænger af hvor stor måling er (✓)
-

Supportability

1. Softwaren 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 Koden indeholder et data-lag, et logik-lag og et GUI-lag ✓
-

Tabel 6.8: Accepttest af Ikke-funktionelle krav

Bilag

Fejlrappor

Logbog

Mødereferart

Kode

Samarbejdsaftale