



AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI  
3. SEMESTERPROJEKT

---

# Dokumentation

---

*Gruppe 2*

Anne Bundgaard Hoelgaard (201404492)  
Mette Hammer Nielsen-Kudsk (201408391)  
Ditte Heebøll Callesen (201408392)  
Martin Banasik (201408398)  
Albert Jakob Fredshavn (201408425)  
Johan Mathias Munk (201408450)

*Vejleder*

Studentervejleder  
Peter Johansen  
Aarhus Universitet

13. december 2015

*Gruppemedlemmer*

---

Anne Bundgaard Hoelgaard (201404492)

---

Dato

---

Mette Hammer Nielsen-Kudsk (201408391)

---

Dato

---

Ditte Heebøll Callesen (201408392)

---

Dato

---

Martin Banasik (201408398)

---

Dato

---

Albert Jakob Fredshavn (201408425)

---

Dato

---

Johan Mathias Munk (201408450)

---

Dato*Vejleder*

---

Peter Johansen

---

Dato

# Ordliste

---

Ord	Forklaring
(F)URPS+	Et akronym, der repræsenterer en model til klassificering af softwarens kvalitet
GUI	Graphical User Interface (Grafisk brugergrænseflade)
VPN	Virtual Private Network
DAQ	Data acquisition, NI-6009 DAQ

# Indholdsfortegnelse

---

<b>Ordliste</b>	<b>ii</b>
<b>Kapitel 1 Indledning</b>	<b>1</b>
<b>Kapitel 2 Kravspecifikation</b>	<b>2</b>
2.1 Versionshistorik . . . . .	2
2.2 Godkendelsesformular . . . . .	3
2.3 Indledning . . . . .	4
2.4 Systembeskrivelse . . . . .	4
2.5 Funktionelle krav . . . . .	4
2.5.1 Aktør-kontekstdiagram . . . . .	4
2.5.2 Aktørbeskrivelse . . . . .	5
2.5.3 Use case-diagram . . . . .	5
2.5.4 Use Cases . . . . .	6
2.6 Ikke-funktionelle krav . . . . .	9
2.6.1 (F)URPS+ . . . . .	9
<b>Kapitel 3 Systemarkitektur</b>	<b>12</b>
3.1 Hardware . . . . .	13
3.1.1 Design . . . . .	13
3.1.2 Implementering . . . . .	14
3.1.3 Modultest . . . . .	18
3.2 Software . . . . .	23
3.2.1 Design . . . . .	23
3.2.2 Implementering . . . . .	29
3.2.3 Modultest . . . . .	37
<b>Kapitel 4 Accepttest</b>	<b>38</b>
4.1 Accepttest af Use Cases . . . . .	38
4.2 Indledning . . . . .	38
4.2.1 Use Case 1 . . . . .	38
4.2.2 Use Case 2 . . . . .	39
4.2.3 Use Case 3 . . . . .	40
4.2.4 Use Case 4 . . . . .	41
4.2.5 Use Case 5 . . . . .	42
4.3 Accepttest af ikke-funktionelle krav . . . . .	43
4.4 Godkendelsesformular . . . . .	49
<b>Figurer</b>	<b>50</b>

# Indledning 1

---

I dag bruges blodtryksmålere mange steder, både på hospitalet og i hjemmet. Blodtryksmålere kan måle en persons blodtryk, hvor den viser puls, samt diastoliske- og systoliske tryk i numerisk form og afbilledet i en graf.

Vi har valgt at arbejde ud fra, at blodtryksmåleren skal bruges til forskning. Derfor skal systemet gemme samtlige målinger, så en forsker senere kan tilgå dem. Samtidig skal puls og tryk vises på en graf, som skal være nem at aflæse. Brugeren vil kunne benytte målere gennem et interface, hvor han kan starte og gemme målinger. Det er også her grafen vises. Der var fra start givet en række krav til systemet, samtidig har gruppen valgt at tilføje nogle flere for at få de ting løst, gruppen synes var vigtigt. Disse kan findes i Kravspecifikationen og under Krav.

Nærmere informationer om opbygning af hardware og software kan findes under Systemarkitektur, som er delt ind efter Hardware og Software. Her under findes også Modultest. Under Modultest kan det læses, hvordan vi har testet systemet samlet og enkelt hvis for hardware og software. Under Accepttest ses det, om systemet opfylder kravene der blev sat.

## Ansvarsområde

### Initialer:

Albert Jakob Fredshavn - AJF

Martin Banasik - MBA

Mette Hammer Nielsen-Kudsk - MHNK

Ditte Heebøll Callesen - DHC

Johan Mathias Munk - JMM

Anne Bundgaard Hoelgaard - ABH

Afsnit	Ansvarlig
--------	-----------

---

Indledning	
------------	--

Kravspecifikation	
-------------------	--

Hardware arkitektur	
---------------------	--

Software arkitektur	
---------------------	--

Software implementering	
-------------------------	--

Accepttest	
------------	--

Fejlrapport	
-------------	--

# Kravspecifikation 2

---

## 2.1 Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse
0.1	21-09-2015	MH NK og MB A	Oprettelse og udfyldning af kravspecifikation
0.2	24-09-2015	DHC og ABH	Omskrivning af UC1 - UC5
0.3	28-09-2015	ABH	Ikke-funktionelle krav
0.4	08-10-2015	Alle	Tilrette efter review med Grp. 1
0.5	15-10-2015	MB A	Indskrevet i LaTeX
0.6	11-11-2015	ABH	Ændre Use Case 1 og 2 efter review med Grp. 4
0.7	20-10-2015	MH NK	Tilretning
0.8	26-11-2015	MH NK	Retning af hele kravspec.
0.9	09-12-2015	DHC	Rettelser ift. software
1.0	09-12-2015	MH NK	Rettelse af afsnit i rapport og dokumentation
1.1	10-12-2015	DHC, ABH	Rettelser i forhold til slutprodukt

## 2.2 Godkendelsesformular

Forfattere      Anne Hoelgaard, Ditte Heebøll Callesen, Martin Banasik, Albert Fredshavn, Mathias Munk og Mette Hammer Nielsen-Kudsk

---

Godkendes af   Peter Johansen

Antal sider     50

Kunde            IHA

Ved underskrivelse af dette dokument accepteres det af begge parter, som værende kravene til udviklingen af det ønskede system.

---

Sted

---

Dato

---

Kundens underskrift

---

Leverandørens underskrift

## 2.3 Indledning

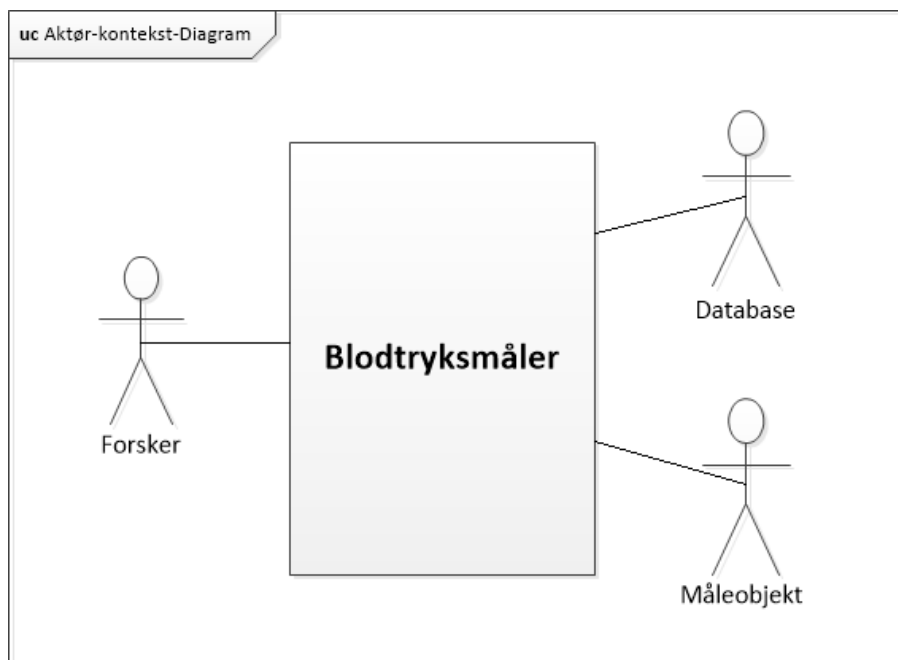
På baggrund af krav fra kunden samt, hvad leverandøren finder muligt, er denne kravspecifikation blevet udarbejdet. Kravspecifikation har til formål at specificere kravene til produktet. Dette projekt tager udgangspunkt i en blodtryksmåler, hvortil der er en række aktører, som interagerer med systemet, der er beskrevet yderligere nedenfor.

## 2.4 Systembeskrivelse

Blodtryksmålersystemet ønskes udviklet således at systolisk og diastolisk blodtryk samt puls kan bestemmes ud fra en invasiv arteriel blodtryksmåling. Der udvikles instrumentering til den udleverede transducer som hardware og et software-program til kontinuerligt visning af målt blodtryk, samt til udskrivelse af løbende systoliske, diastoliske og puls værdier. Disse to dele udgør til sammen systemet.

## 2.5 Funktionelle krav

### 2.5.1 Aktør-kontekstdiagram



Figur 2.1: Aktør-kontekstdiagram

På figur 2.1 ses aktørerne til at være: Forsker, Måleobjekt og Database. Herunder er der en detaljeret beskrivelse af hver aktør.

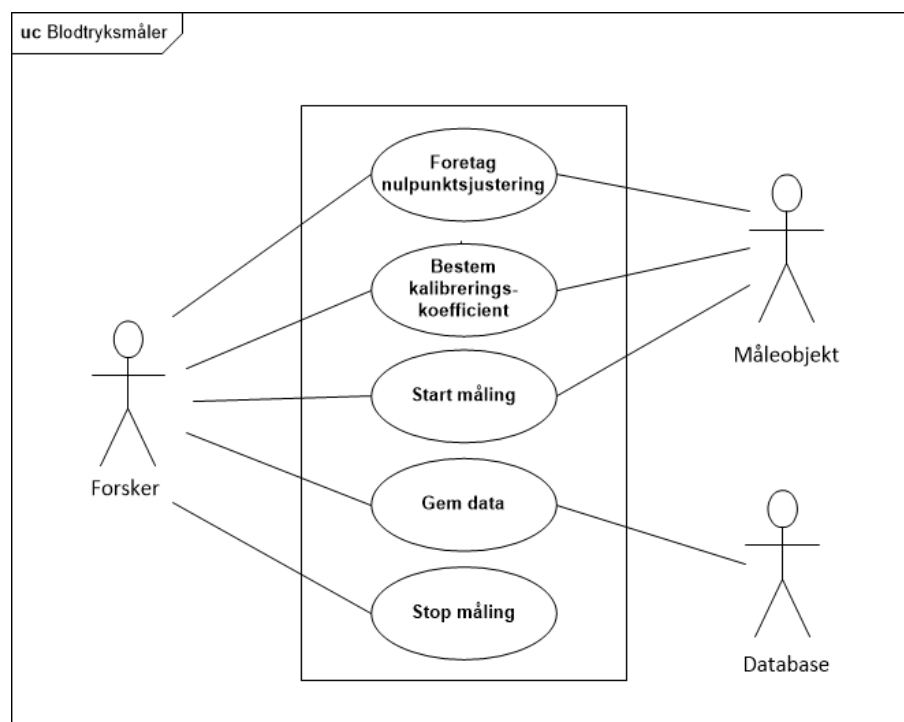


### 2.5.2 Aktørbeskrivelse

Aktørnavn	Type	Beskrivelse
Forsker	Primær	Forskeren er aktøren, der starter måling, giver besked om at data ønskes gemt, navngiver målingen samt afslutter måling af blodtryk
Database	Sekundær	Databasen er aktøren, hvori måledata bliver gemt
Måleobjekt	Sekundær	Måleobjekt er aktøren, hvorfra blodtrykssignalet indhentes. Måleobjektet er tilkoblet transduceren I den endelige version er måleobjektet In Vitro maskinen, som findes i Cave Lab Under løbende test i udviklingsprocessen benyttes Analog Discovery og Waveform

Tabel 2.3: Aktørbeskrivelse

### 2.5.3 Use case-diagram



Figur 2.2: Use Case-diagram

Diagrammet ovenfor viser systemets fem Use Cases: Foretag nulpunktsjustering, Bestem kalibreringskoefficient, Start måling, Gem data og Stop måling. Herunder følger en nærmere beskrivelse af de enkelte Use Cases, gennem et fully-dressed Use Case skema.

Systemet består af en computer, hvor softwaren er placeret, en NI-DAQ, en Analog Discovery samt en transducer med tilhørende implementering. Systemet gør det muligt at foretage en blodtryksmåling på et måleobjekt, som er tilsluttet og sender disse signaldata

ind i systemet via transduceren og det tilhørende hardware, hvor signalet vises. Det ønskede interval af blodtrykssignalet gemmes så i databasen.

I softwaren benyttes algoritmer til at analysere signalet, ud fra opsatte grænseværdier, så systolisk, diastolisk og puls værdier hentes frem. Disse algoritmer undersøger signalet for, hvor signalets bølgetoppe og -bunde er placeret. Da top er signalets systoliske værdi og bund er signalets diastoliske. Puls bestemmes ved at tælle antallet af blodtryksperioder pr. minut.

Brugergrænseflade er det, som forsker initierer med, altså hvorfra systemet aktiveres. Brugergrænsefladen forkortes til GUI.

#### 2.5.4 Use Cases

##### Use Case 1

Scenarie	Hovedscenarie
Navn	Foretag nulpunktsjustering
Mål	At få foretaget en nulpunktsjustering
Initiering	Startes af Forsker
Aktører	Forsker (primær), Måleobjekt (sekundær)
Referencer	
Samtidige forekomster	Én nulpunktsjustering pr. kørsel
Forudsætninger	Alle systemer er ledige og operationelle
Resultat	Nulpunktsjustering er blevet foretaget efter ønske
Hovedscenarie	<ol style="list-style-type: none"> <li>1. Pop-up vindue for nulpunktsjustering er åbent</li> <li>2. Forsker trykker på Foretag-knap:</li> <li>3. Systemet foretager nulpunktsjustering og vinduet lukker ned.</li> </ol>
Undtagelser	-

*Tabel 2.4: Fully dressed Use Case 1*

##### Use Case 2

Scenarie	Hovedscenarie
Navn	Bestem kalibreringskoefficient
Mål	At få bestemt kalibreringskoefficienten
Initiering	Startes af Forsker

Aktører	Forsker (primær)
Referencer	Ingen
Samtidige forekomster	Én kalibrering pr. måling
Forudsætninger	Alle systemer er ledige og operationelle. Væskesøjle og computer med en WaveForm er tilgængeligt
Resultat	Kalibreringskoefficient er blevet indtastet i XML-fil
Hovedscenarie	<ol style="list-style-type: none"> <li>1. Forsker tilslutter WaveForm og væskesøjle ved 50 mmHg til systemets hardware.</li> <li>2. Output spænding fra hardware aflæses i WaveForm</li> <li>3. Beregning foretages</li> <li>4. Forsker indtaster beregnet kalibreringskoefficient i konfigurations XML-fil</li> <li>5. Kalibreringskoefficienten kan tilgås af systemet</li> </ol>
Undtagelser	-

*Tabel 2.5: Fully dressed Use Case 2***Use Case 3**

Scenarie	Hovedscenarie
Navn	Start Måling
Mål	At få foretaget en blodtryksmåling
Initiering	Startes af Forsker
Aktører	Forsker (primær), Måleobjekt (sekundær)
Referencer	Use Case 1
Samtidige forekomster	Ét signal pr. måling
Forudsætninger	Use Case 1 er kørt succesfuldt, samt alle systemer kører og er klar til at foretage en måling
Resultat	Systolisk-, diastolisk blodtryk, puls og blodtryksgraf bliver vist på GUI
Hovedscenarie	<ol style="list-style-type: none"> <li>1. Forsker indtaster Forsøgsnavn</li> <li>2. Filteret signal er valgt per default af systemet</li> </ol>

3. Forsker trykker på Start-knap på GUI
4. Signal for blodtryk vises på GUI
5. Systolisk og diastolisk blodtryk samt puls bliver vist i bokse på GUI

[*Udvidelse 1:*] Forsker vælger filtreret/ufiltreret signal

---

Undtagelser og Udvidelser	[ <i>Udvidelse 1</i> ] Forsker vælger filtreret/ufiltreret signal
a.	Forsker vælger ufiltreret signal
b.	Det viste signal er nu ufiltreret
c.	Forsker vælger filtreret signal
d.	Det viste signal er nu filtreret

---

*Tabel 2.6: Fully dressed Use Case 3*

## Use Case 4

---

Scenarie	Hovedscenarie
Navn	Gem data
Mål	At gemme data i databasen
Initiering	Startes af Forsker
Aktører	Forsker (primær), Database(sekundær)
Referencer	Use Case 1 og Use Case 3
Samtidige forekomster	Ét signal pr. måling
Forudsætninger	Use Case 1 er kørt succesfuldt, Use Case 3 kører. VPN er tilsluttet
Resultat	Signalets rådata er blevet gemt i en Database under Forsøgsnavn og et autogenereret Id

---

Hovedscenarie	<ol style="list-style-type: none"> <li>1. Forsker trykker på Start Gem-knap</li> <li>2. Systemet gemmer det fremadrettede signals rådata i Databasen</li> <li>3. Forsker trykker på Stop Gem-knap for at stoppe med at gemme</li> </ol>
	[ <i>Undtagelse 1:</i> ] Forsker trykker på Stop Måling-knap

4. Det vises at data er gemt ved at filnavnet(Forsøgsnavn og Id) for målingen vises på GUI.

---

Undtagelser	[ <i>Undtagelse 1:</i> ] Forsker trykker på Stop Måling-knap
a.	Systemet gemmer ikke målingen og blodtryksgrafen fastholdes.

---

Tabel 2.7: Fully dressed Use Case 4

## Use Case 5

---

Scenarie	Hovedscenarie
Navn	Stop måling
Mål	At stoppe målingen af blodtryk
Initiering	Startes af Forsker
Aktører	Forsker (primær)
Referencer	Use Case 1 og 3
Samtidige forekomster	Ét signal pr. måling
Forudsætninger	Use Case 1 er kørt succesfuldt, Use Case 3 kører
Resultat	Måling af blevet stoppet
Hovedscenarie	<ol style="list-style-type: none"> <li>1. Forsker trykker på Stop Måling-knap</li> <li>2. Målingen stopper og blodtryksgrafen fastholdes.</li> </ol>
Undtagelser	-

---

Tabel 2.8: Fully dressed Use Case 5

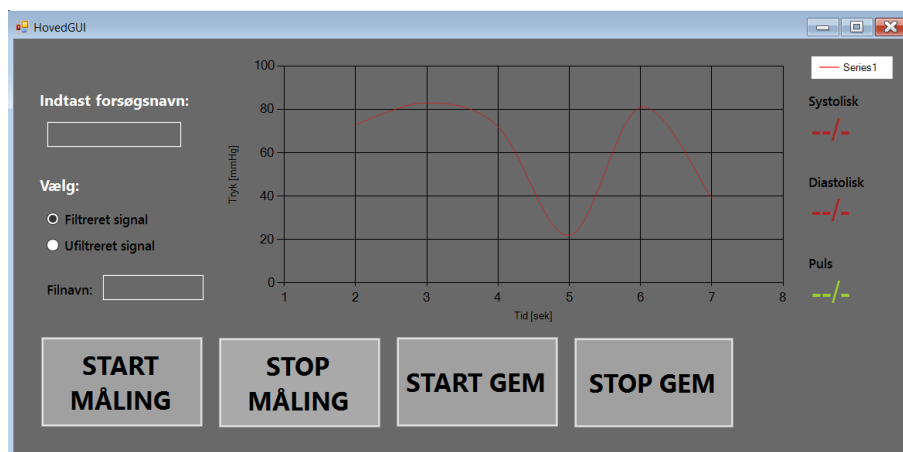
## 2.6 Ikke-funktionelle krav

### 2.6.1 (F)URPS+

#### Functionality

1. Blodtryksmåleren skal indeholde en Start Måling-knap til at igangsætte målingerne.
2. Blodtryksmåleren skal indeholde en Stop Måling-knap, hvorfra måling kan stoppes.

3. Blodtryksmåleren skal indeholde en Start Gem-knap til påbegyndelse af at gemme måling i Database.
4. Blodtryksmåleren skal indeholde en Stop Gem-knap til afslutning af at gemme måling i Database.
5. Blodtryksmåleren skal indeholde en tekstboks til forsøgsnavn, hvori forsker indtaster det pågældende forsøgsnavn.
6. Blodtryksmåleren skal indeholde radiobutton til filtreret signal, denne skal være default valget.
7. Blodtryksmåleren skal indeholde radiobutton til ufiltreret signal.
8. Blodtryksmåleren skal indeholde tekstbokse til puls, systolisk og diastolisk blodtryk som vises med op til tre cifre.
9. Blodtryksmåleren skal indeholde en tekstboks som viser filnavn(forsøgsnavn og id) på målingen, efter måling er gemt.
10. GUI'en skal se ud som vist på figur 2.3:



Figur 2.3: Skitse af GUI

## Usability

1. Forskeren skal kunne starte en default-måling maksimalt 30 sekunder efter systemet er startet.

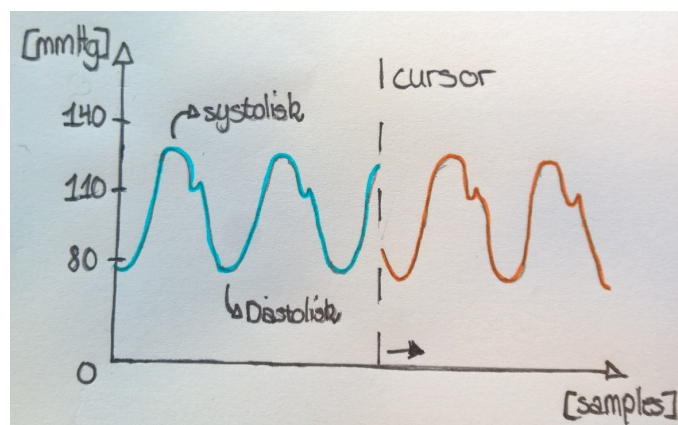
## Reliability

1. Det skal maksimalt tage 5 timer at gendanne systemet (MTTR - Mean Time To Restore)
2. Systemet skal have en opetid uden nedbrud på minimum 1 måned (720 timer) (MTBF - Mean Time Between Failure).
3. Systemet skal have en opetid/køretid på:

$$Availability = \frac{MTBF}{MTBF + MTTR} \cdot 100 = \frac{720}{720 + 5} \cdot 100 = 99,31\% \quad (2.1)$$

### Performance

1. Blodtryksmåleren skal, indenfor 3 sekunder, kunne vise systolisk og diastolisk blodtryk via graf. Dette accepteres med en tolerance på +/- 15 %.
2. Blodtryksmåleren skal, indenfor 5 sekunder fra der er trykket på Stop Gem-knap, have gemt målingerne i Databasen. Dette accepteres med en tolerance på +/- 15 %.
3. Grafen vises i ét vindue, hvor y-aksen måles i mmHg (millimeter kviksølv) og x-aksen i tid i sekunder.
4. Hvert 3. sekund skal værdier for systolisk og diastolisk blodtryk samt puls opdateres. Dette accepteres med en tolerance på +/- 15 %.
5. Graf for blodtryk skal køre kontinuerligt i GUI efter følgende princip(figur 2.4), hvor det blå signal erstatter det orange signal ved, at den seneste måling altid sættes ved cursorens placering.



Figur 2.4: Graf for blodtryk

6. Når der trykkes på Stop Gem-knap gemmes signals rådata under det indtastede forsøgsnavn og et autogenereret id. "*forsøgsnavn\_id*".
7. Systemet skal kunne måle blodtryksværdier fra 0 til 250 mmHg.

### Supportability

1. Forskeren skal kunne udskifte batterierne til hardwaren på 2 minutter.
2. Softwaren skal opbygges med lav kobling.

# Systemarkitektur 3

---

Version	Dato	Ansvarlig	Beskrivelse
0.1	03-11-2015	MBA	Oprettelse
0.2	10-11-2015	DHC, MBA	HW Start af skrivning, indsætning af billeder
0.3	10-11-2015	ABH	SW Start på design, indsætning af diagrammer
0.4	11-11-2015	DHC	HW Design Forstrækning
0.5	13-11-2015	ABH	SW Design klasse- og metodeidentifikation
0.6	18-11-2015	ABH	HW Rettelse af diagrammer
0.7	18-11-2015	DHC, AJF	HW Implementering Forstrækning, Modultest Lavpas
0.8	18-11-2015	MHNK, JMM	SW Design, Rettelse af domænemodel
0.9	18-11-2015	ABH	SW Design, Mere metodeidentifikation
1.0	20-11-2015	MHNK	SW Indskrivning af alle sekvensdiagrammer
1.1	26-11-2015	DHC	HW Modultest, Kalibrering ved vandsøjle
1.2	26-11-2015	DHC, AJF	HW Design Lavpas
1.3	02-12-2015	DHC	HW Referencer
1.4	02-12-2015	MHNK	HW Rettelser i tekst
1.5	02-12-2015	DHC, MBA	HW Modultest
1.6	04-12-2015	ABH	SW Implementering, Generelt, Analyse og Digitalt filter
1.7	06-12-2015	ABH	SW Implementering, Kalibrering og nulpunktsjustering
1.8	09-12-2015	DHC	Rettelser i tekst
1.9	09-12-2015	ABH, JMM	SW Implementering Observer-Strategy, Analyse og Digital Filter

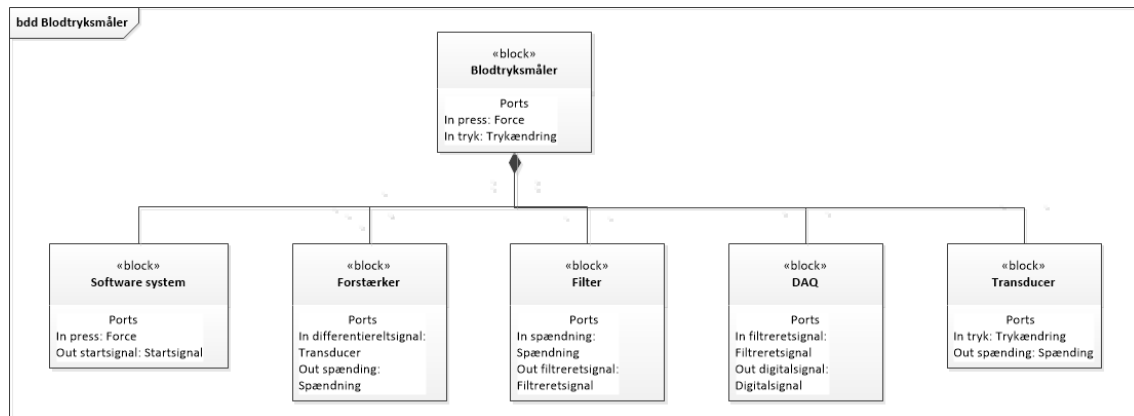
I det følgende beskrives arkitekturen for systemet. Systemarkitekturen er vores udviklingsramme for den videreudvikling af design og implementering af blodtrykssystemet. Designet af systemet er grebet an således at, der først kigges på det overordnede system, hvorefter systemet arbejdes ned i mindre brudstykker. Dette gøres ved at benytte diagrammer med tilhørende beskrivelser.



## 3.1 Hardware

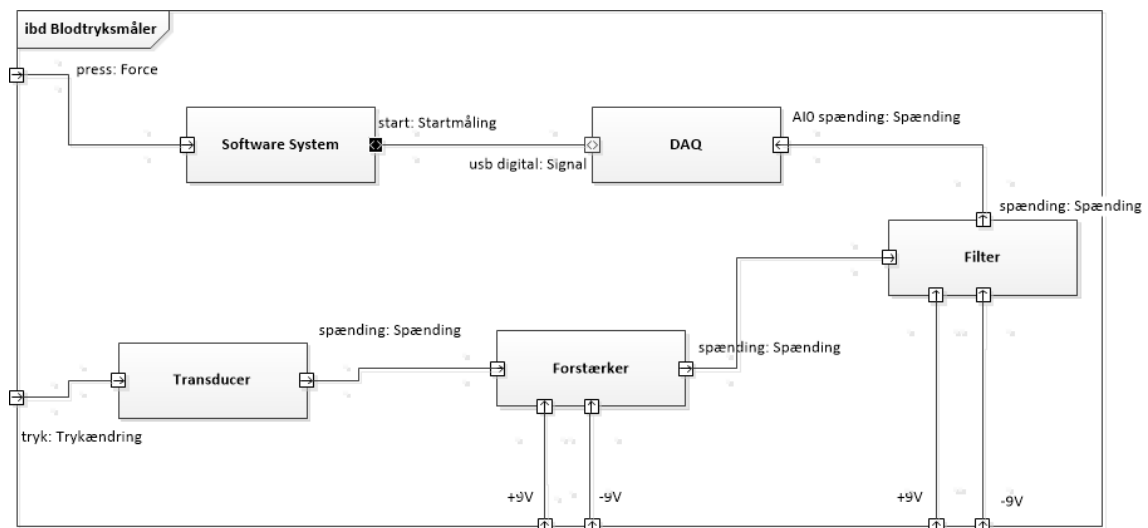
### 3.1.1 Design

Systemets hardware kan illustreres i et BDD. Det ses på figur 3.1 at systemet består af fem hardware blokke: software system, forstærker, filter, DAQ og transducer. Disse fem blokke udgør til sammen selve blodtryksmåleren.



Figur 3.1: Block Definition Diagram for hardware

Ovenstående BDD-diagram fører videre til udarbejdelsen af IBD for hardware komponenterne. I IBD diagrammet vises koblingen mellem de forskellige blokke gennem port forbindelser. Det ses at signalet starter ved transduceren, hvorefter det bliver behandlet gennem forstærker, filter og DAQ. Til sidst sendes det ind i software systemet, som bliver påvirket af tryk på knapper på GUI.



Figur 3.2: Internal Block Diagram for hardware

### Forstærkning

Transduceren måler en trykændring som den omsætter til en spænding. Dette er udtrykt ved et differentieret signal, som sendes ind i forstærker-blokken. Da signalet

fra transduceren er en lav spænding, skal det forstærkes op, for at passe med DAQ'ens input. Denne forstærkning udregnes ud fra det maksimale output fra transduceren og det maksimale input til DAQ'en. Se beregningerne under Implementering.

Under simulering bruges Analog Discovery som en funktionsgenerator, der simulerer det differentieret signal. Analog Discovery har en usikkerhed, når der arbejdes med små spændinger. Dette kan modarbejdes vha. spændingsdeler princippet. Dette gør at Analog Discovery kan sende en højere spænding ind i systemet, så usikkerheden mindskes. Dette bruges kun under simulering og teste af hardwaren.

## Lavpas

I projektet skal der laves et 2. ordens lavpasfilter. Filteret skal laves for at sikre, at der ikke opstår aliasering.

Aliasering [?] er, hvor signalet bliver gentaget. Når man har signalet i det digitale domæne, bliver spektret for signalet en periodisk funktion. Det vil sige, at den gentager sig selv, efter et bestemt stykke tid.

Det skal sikres, at der ikke kommer overlap mellem signalet og et alias. Da det ellers kunne give anledning til misforståelser. Derfor laves et lavpasfilter, som sikrer at der ikke ligger noget signal ved den halve samplingsfrekvens. Signalet her kan med fordel gøres så lille at DAQ'en ikke kan læse det, dvs. signalet skal være mindre end  $1/2 \cdot LSB$  (Least Significant Bit).

Lavpasfilteret skal være et Sallen-Key Butterworth-filter med en knæfrekvens på 50 Hz og en samplingsfrekvens på 1 kHz. Ud fra oplysninger givet til projektet, vides det at filteret skal dæmpe signalet med 20 dB, under antagelse af at, den forekommende støj er mindre end signalet, også når støjen forekommer over knæfrekvensen.

Ved en typisk blodtryksmåling forekommer der ikke signal over 50 Hz, samtidigt er signalet her aftaget med ca. 70 dB. For at få signalet, ved den halve samplingsfrekvens til at være  $1/2 \cdot LSB$ , skal det ydeligere dæmpes 20 dB. Derfor oplyses filterets til at være 50 Hz, da dette giver en minimums dæmpning på 20 dB pr. dekade.

### 3.1.2 Implementering

#### Forstærkning

For at få den rette forstærkning er det blevet valgt, at benytte instrumentationsforstærkeren INA-114. Her kan transduceren sættes på med det differentierede signal. INA114 er valgt da følgende gælder[?] for instrumentationsforstærkere:

- Differentielt input - single ended output
- Gain justering med ændring af kun én modstand
- Meget høj indgangsimpedans
- Stor Common Mode Rejection Ratio(CMRR)

Under opbygning og modultestning vil det differentierede signal blive simuleret af Analog Discovery.

For at udregne den korrekte forstærkning, bruges følsomheden fra transduceren og eksistationsspændingen. Først udregnes det maksimale output fra transduceren:

$$9V \cdot 250mmHg \cdot 5\mu \cdot 10^{-5}uV/V/mmHg = 11.25mV \quad (3.1)$$

Da det er besluttet at det maksimale input til DAQ'en [?] er 5V, kan forstærkningen (Gain) nu udregnes:

$$\begin{aligned} 5V &= 11.25mV \cdot G \\ G &= 444.44 \end{aligned} \quad (3.2)$$

[?] For at få den rette forstærkning udregnes den eksterne modstand ( $R_g$ ) til INA114. INA114's forstærkning afhænger af størrelsen på  $R_g$ , hvis modstanden er stor, er forstærkningen lille og omvendt.  $R_g$  udregnes ved formlen:

$$\begin{aligned} G &= 1 + \frac{50k\Omega}{R_g} \\ 444.44 &= 1 + \frac{50k\Omega}{R_g} \Rightarrow R_g = 112.75\Omega \end{aligned} \quad (3.3)$$

Derved fås en værdi for den eksterne modstand til INA114, som skaber den ønskede forstærkning.

Det skal nu sikres at dette kan lade sig gøre. Derfor sikres det, at den ønskede forstærkning kan ske ved båndbredden. Dette kan undersøges da produktet af forstærkning og båndbredde er en konstant. Konstanten aflæses i databladet for INA114[?].

$$\begin{aligned} 1000000Hz &= G \cdot BW \\ BW &= 2250Hz \end{aligned} \quad (3.4)$$

Da båndbredden ligger over knækfrekvensen for lavpas filtret, er dette godkendt. Hvis båndbredde havde ligget under knækfrekvensen vil operationsforstærkeren ikke have kunnet arbejde med de ønskede frekvenserne. Derfor er det vigtigt at båndbredden er bred nok til at kunne indeholde frekvenser fra begge side af knækfrekvensen.

For at imødekomme usikkerheden ved Analog Discovery med lave spændinger, laves et kredsløb efter spændingsdelerprincippet. Signalerne fra Analog Discovery skal sendes igennem dette kredsløb, hvor de efter spændingsdelerprincippet gøres mindre. I kredsløbet benyttes to modstande, hvis værdier er  $R_1 = 100k\Omega$  og  $R_2 = 1k\Omega$ . Da vi kender signalet som skal ind i INA114 og modstandene i kredsløbet, kan størrelsen af den spænding, som skal sendes fra Analog Discovery, findes:

$$\begin{aligned} U_{INA} &= U_{analog} \cdot \frac{R_2}{R_1 + R_2} \\ 11.25mV &= U_{analog} \cdot \frac{1k\Omega}{100k\Omega + 1k\Omega} \Rightarrow U_{analog} = 1.1362V \end{aligned} \quad (3.5)$$

Derved kan Analog Discovery sende signaler med en højere spænding ud og usikkerheden mindskes. Der er taget højde for at, hvis modstandene i kredsløbet bliver for store, vil det skabe en termisk usikkerhed. Derfor er modstandene valgt som de er. Dette er kun under simulering, når transduceren benyttes, bruges spændingsdeleren ikke.

### Lavpas

For at opnå den ønskede effekt i lavpasfilteret, blev det oplyst at  $f_c = 50$  Hz,  $f_s = 1$  kHz,  $R_1 = R_2$  og  $C_2 = 680$  nF. Ud fra disse værdier, udregnes de resterende komponentværdier for filteret.

Overføringsfunktionen for et 2. ordens filter er:

$$H(z) = \frac{\omega_n^2}{(s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2)} \quad (3.6)$$

For at finde overføringsfunktionen for det gældende system, vides det at følge ligninger gælder [? ]:

$$\begin{aligned} \omega_n &= 2 \cdot \pi \cdot 50 = \frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \\ 2 \cdot \zeta \cdot \omega_n &= \frac{1}{C_2} \cdot \left( \frac{R_1 + R_2}{R_1 \cdot R_2} \right) \end{aligned} \quad (3.7)$$

Derved fås en overføringsfunktion som hedder:

$$H(z) = \frac{\left( \frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \right)^2}{s^2 + \left( \frac{1}{C_2} \cdot \left( \frac{R_1 + R_2}{R_1 \cdot R_2} \right) \cdot s \right) + \left( \frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \right)^2} \quad (3.8)$$

Da det bliver oplyst at  $R_1 = R_2$ , kan funktionen reduceres. Den kan samtidig simplificeres. I sidste ende fås overføringsfunktionen, se Beregninger til overføringsfunktion under Bilag for nærmere udregninger:

$$H(z) = \frac{\frac{1}{C_1 \cdot C_2 \cdot R^2}}{s^2 + s \cdot \frac{1}{R \cdot C_2} + \frac{1}{C_1 \cdot C_2 \cdot R^2}} \quad (3.9)$$

Da der arbejdes med at 2. ordens Butterworth filter, vides det at udsvinget  $\zeta$  skal være 0.7 [? ]. Den sidste overføringsfunktion sammenlignes med den generelle for 2. ordens systemer. Det gælder at  $C_2 = 680 \cdot 10^{-9}$  nF. Det er muligt at isolerer forskellige led. Først isoleres for modstanden:

$$\begin{aligned} \frac{2}{R \cdot C_2} &= 2 \cdot \zeta \cdot \omega_n \\ \frac{2}{R \cdot 680 \cdot 10^{-9}} &= 2 \cdot 0.7 \cdot (2 \cdot \pi \cdot 50) \\ &\Downarrow \\ R &= 6687 \Omega \end{aligned} \quad (3.10)$$

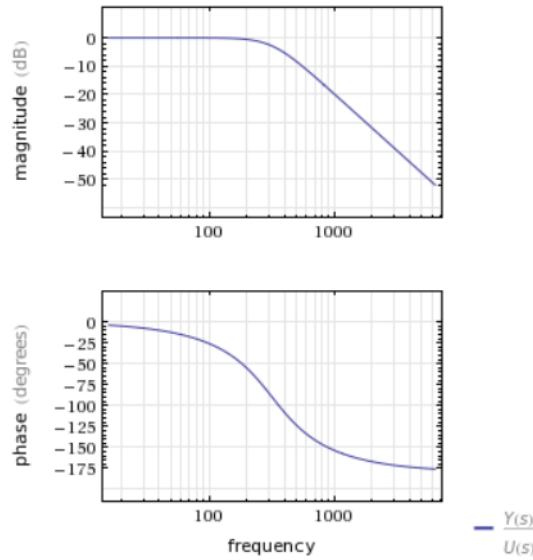
Derved er modstandene udregnet til  $R = 6687 \Omega$ . Nu kan der isoleres for kondensator  $C_1$ :

$$\begin{aligned} \frac{1}{C_1 \cdot C_2 \cdot R^2} &= \omega^2 \\ \frac{1}{C_1 \cdot 680 \cdot 10^{-9} \cdot 6687^2} &= (2 \cdot \pi \cdot 50)^2 \\ &\Downarrow \\ C_1 &= 333 \cdot 10^{-9} \text{ nF} \end{aligned} \quad (3.11)$$

Dette betyder, at  $C1 = 333 \cdot 10^{-9}nF$  og  $C2 = 680 \cdot 10^{-9}nF$ . Derved er alle komponentværdierne til lavpasfilteret fundet og det kan nu realiseres.

Under udviklingen af lavpasfilteret er komponent størrelserne, blevet ændret for at kunne realisere det. De brugte komponent størrelser er:  $R = 6.6k\Omega$ ,  $C1 = 330 \cdot 10^{-9}nF$  og  $C2 = 680 \cdot 10^{-9}nF$ . For at være sikker på at filteret har de ønskede karakteristika, laves et bodeplot for den endelig overføringsfunktion:

$$H(z) = \frac{62500000000}{610929 \cdot \left( s^2 + \frac{250000}{561} \cdot s + \frac{62500000000}{610929} \right)} \quad (3.12)$$

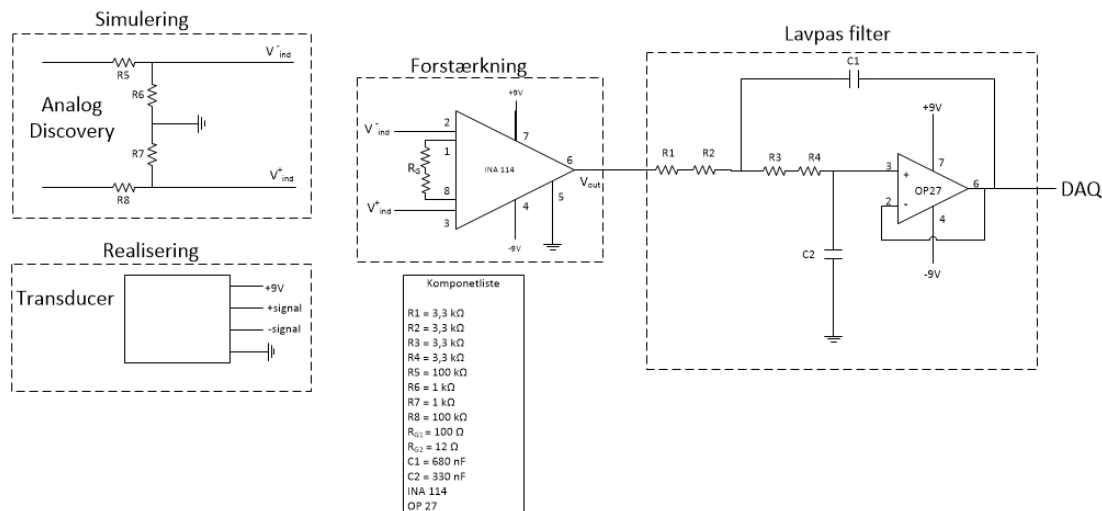


Figur 3.3: Bodeplot

Udregning af den præcise oversving  $\zeta$  ud fra de benyttet komponentværdier:

$$\begin{aligned} \frac{2}{R \cdot C1} &= 2 \cdot \zeta \cdot \omega_n \\ \frac{2}{6600 \cdot 680 \cdot 10^{-9}} &= 2 \cdot \zeta \cdot (2 \cdot \pi \cdot 50) \\ &\Downarrow \\ \zeta &= 0.709 \end{aligned} \quad (3.13)$$

Dvs. de små ændringer i komponent værdierne ikke har haft betydende indflydelse på værdien for  $\zeta$ .



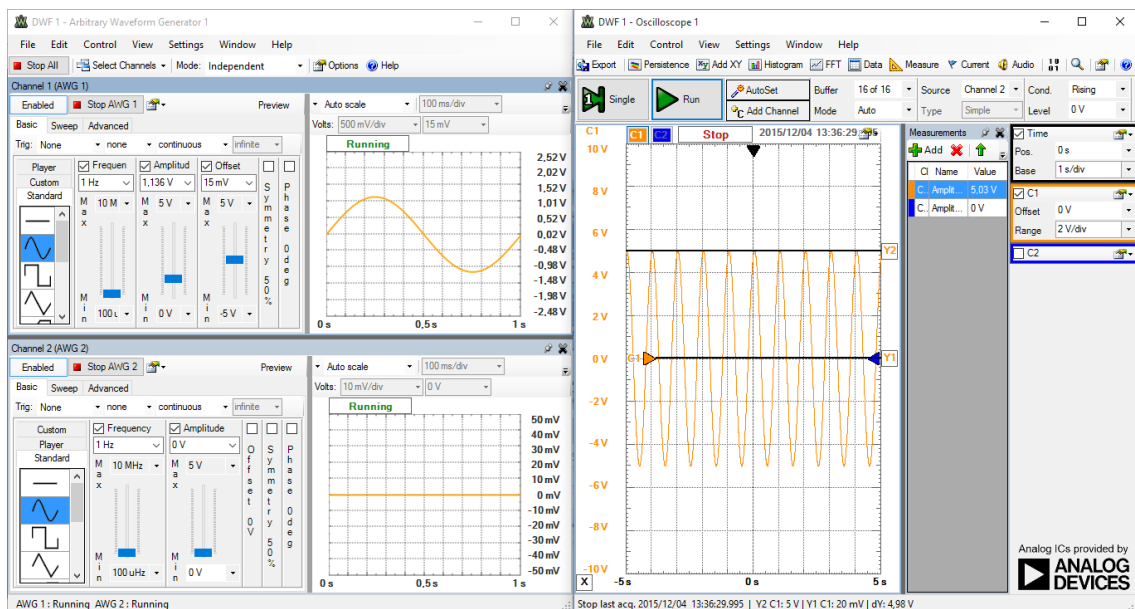
Figur 3.4: Diagram over HW

På figur 3.4 ses et diagram over, hvordan kredsløbet er bygget op. Her ses kredsløbet for realiseringen med transduceren og for simuleringen med Analog Discovery.

### 3.1.3 Modultest

#### Forstærkning

For at teste forstærkningen sendes et differentieret signal ind vha. Analog Discovery. Signalet måles ved udgangen og der ses på, hvor meget signalet er blevet forstærket. På figur 3.5 ses det signal, som sendes ind i forstærknings blokken og det, der måles på udgangen af blokken.



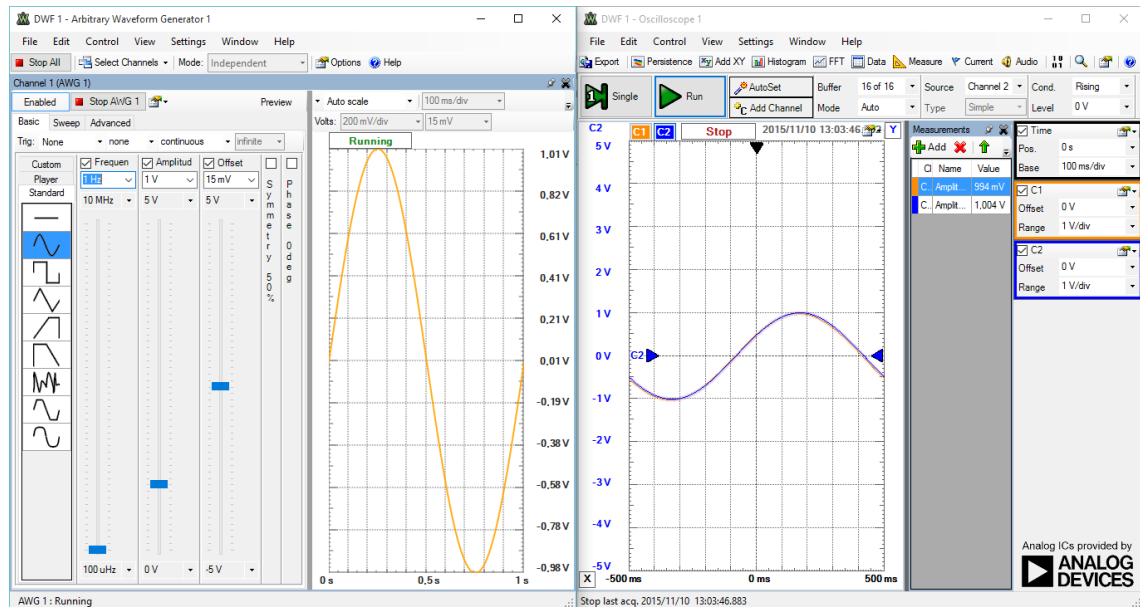
Figur 3.5: Forstærknings blok

På udgangen, ses det at signalet er blevet forstærket op til 5 V DC. Herved er maks. output

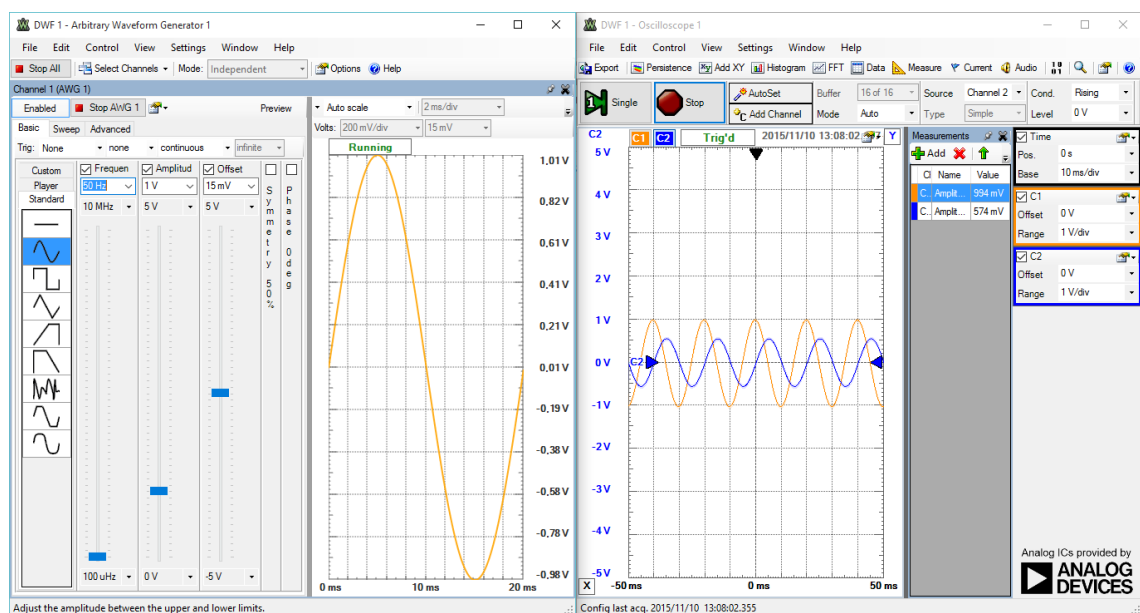
fra transduceren blevet forstærket så det passer med maks. input til DAQ'en. Signalet bliver ikke ændret på andre måde i forstærker blokken.

### Lavpas

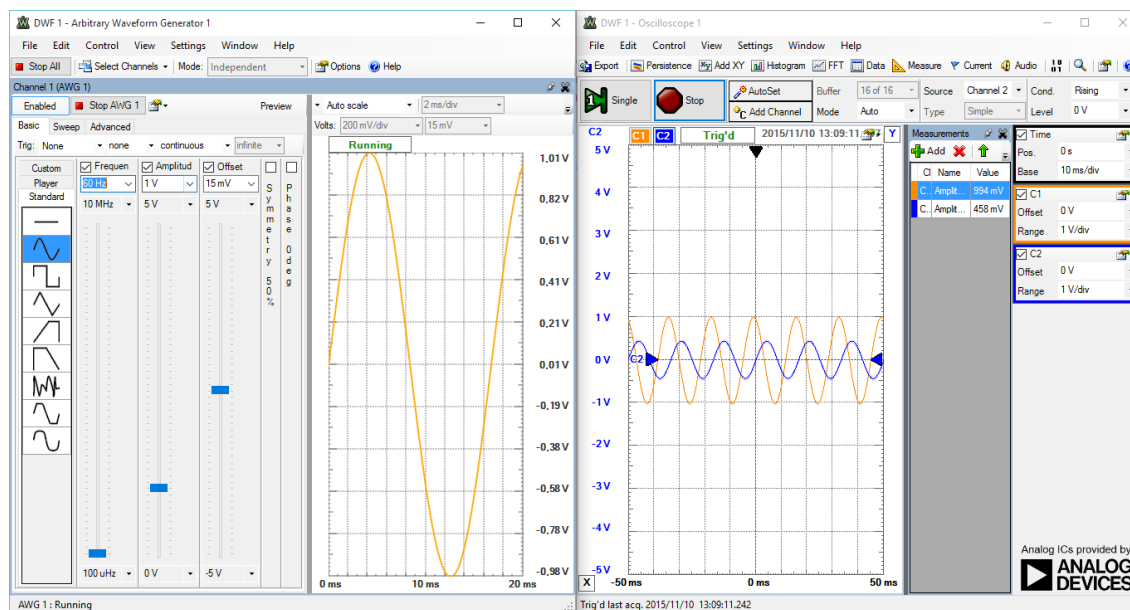
For at teste lavpasfilteret foretages målinger med en sinus, hvor frekvensen variere for hver måling. Fasen aflæses mellem indgang- og udgangssignal. Amplituden aflæses ligeledes for hver måling. Ved knækfrekvensen skal fasedrejningen være  $90^\circ$ . Dette kan aflæses på figur 3.7. Efter knækfrekvensen skal amplituden gå mod nul. Ved målingen for 60 Hz figur 3.8, kan det ses hvordan amplituden er faldet drastisk efter knækfrekvensen.



Figur 3.6: Måling for 10 Hz



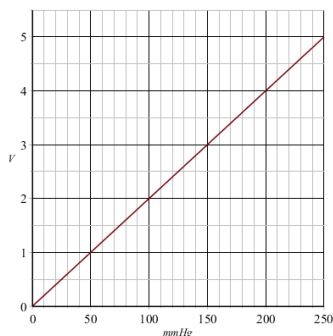
Figur 3.7: Måling for 50 Hz



Figur 3.8: Måling for 60 Hz

### Kalibrering med vandsøjle

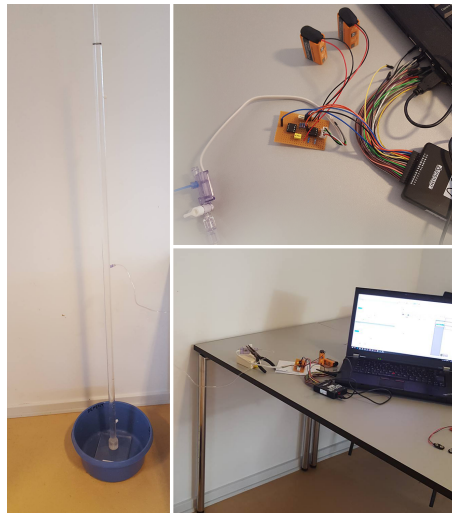
Efter forstærkning og lavpasfilteret er blevet testet hver for sig, udføres en kalibrering af systemet vha. en vandsøjle. Her bruges en udleveret vandsøjle med tre målepunkter, hvor det er angivet, hvor højt trykket(mmHg) er ved hvert af disse punkter. Derved kan det testes om hardware-delen måler den rigtige spænding i forhold til millimeter kviksølv(mmHg). Ud fra det maksimale antal volt (V) spænding og millimeter kviksølv(mmHg) kan det udregnes, hvad hardware skal vise ved 100 mmHg.



Figur 3.9: Graf til kalibrering, fra udregninger

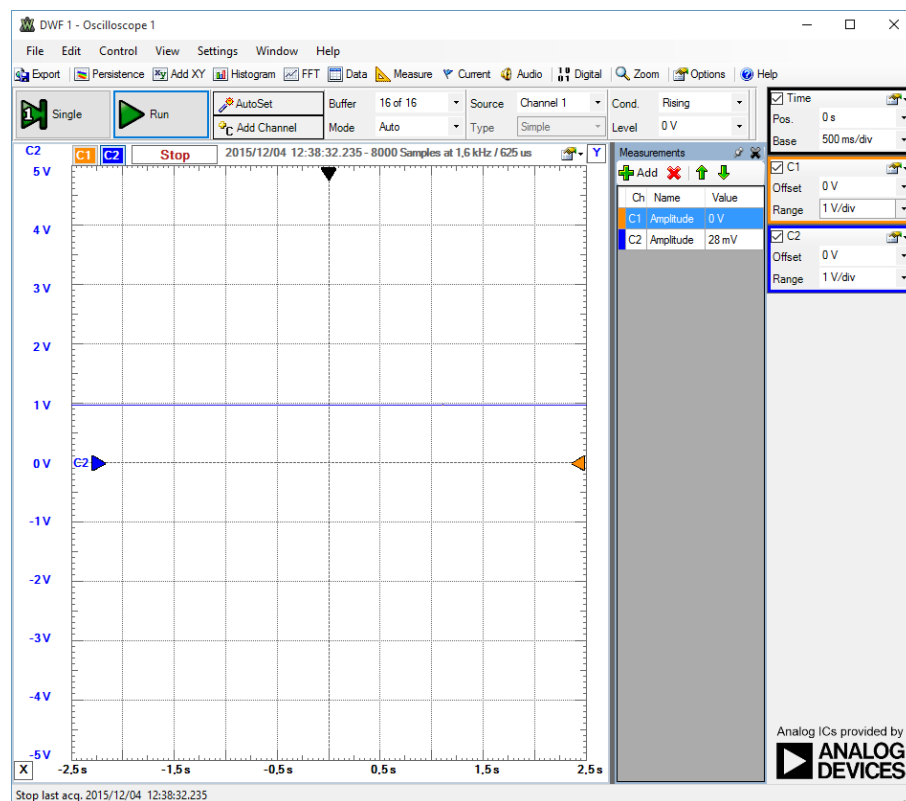
Testen udføres ved, at fylde vand i søjlen til et bestemt punkt. Transducere skal være tilkoblet et af de tre målepunkter, mens de andre er lukket til. Transducere er sat til forstærkningen, der hvor Analog Discovery tidligere har været sat til. Transducere er tilkoblet 0-9V, ved batterierne. På samme måde som ved simuleringen aflæses målingen på computeren ved hjælp af programmet WaveForms. Da det vides, hvilken trykændring der måles på, ved vi fra grafen til kalibreringen, hvilken spænding den skal vise. Dette foretages for de tre målepunkter på vandsøjlen, hvor hver måling sammenlignes med den udregnede graf. For hver måling, skal transducere flyttes til et af de andre målepunkter.

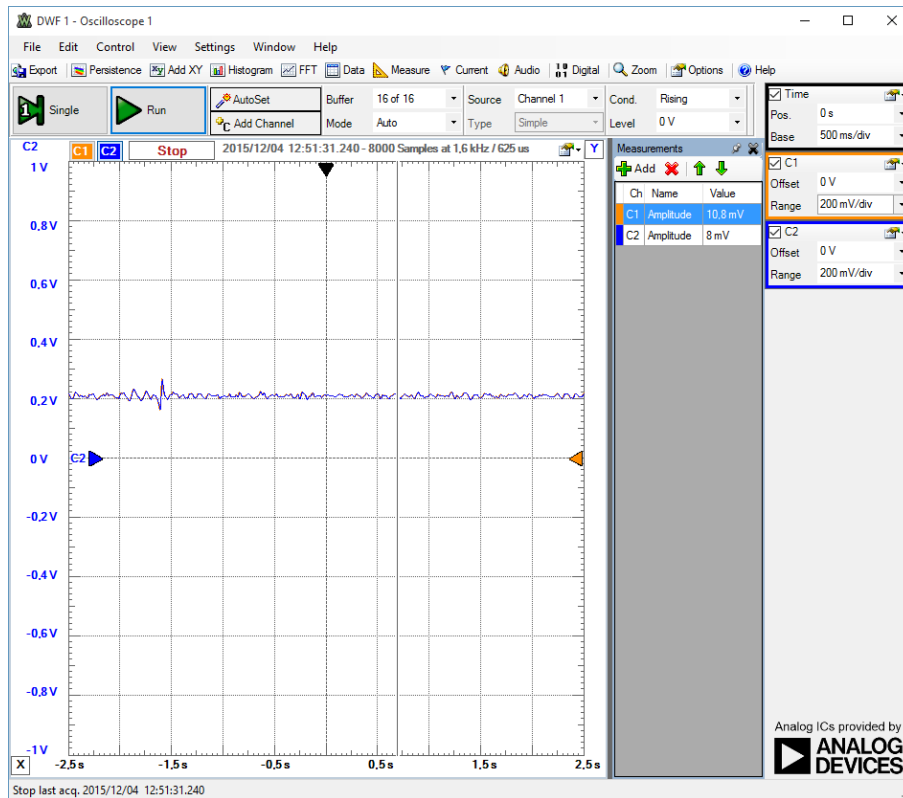


*Figur 3.10: Opstilling*

Opstillingen er gjort klar og der hentes ekstra vand under testen. Vandet skal bruges til at fylde vandsøjlen på til de forskellige målinger.

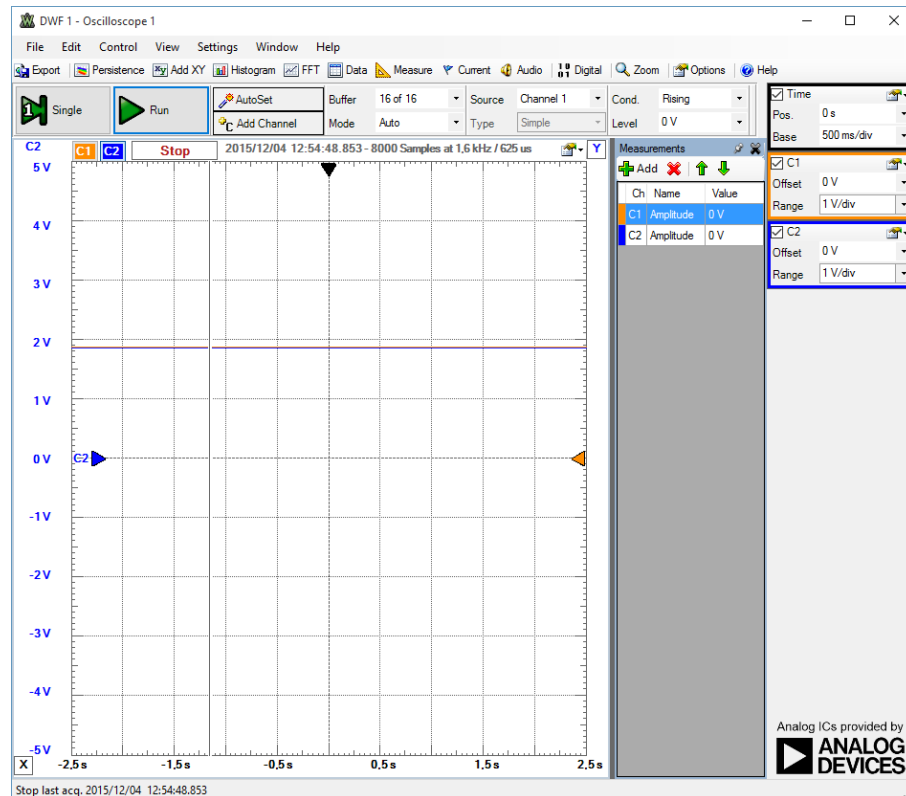
Ud fra grafen i figur 3.9 vides, hvad svaret på hver måling skal være. På figur 3.11 ses målingen, da transduceren var tilkoblet målepunktet for 50 mmHg. Ud fra figur 3.9 vides det at målingen skal vise 1V DC.

*Figur 3.11: Måling ved 50 mmHg*



Figur 3.12: Måling ved 10mmHg

På målingen for 10 mmHg ses en del rystelser(udsving på signalet). Som det ses på figur 3.12 ligger signalet ikke præcist på 0.2V, dette kan skyldes at under testen, skal transduceren være i højde med målepunktet. Pga. korte ledninger, blev det under testen derfor nødvendigt at løfte og holde transducer, VEVO Board og Analog Discovery i højde med målepunktet.



Figur 3.13: Måling ved 100mmHg

Ved målingen for 100mmHg skulle der måles en spænding på 2V. Som det ses på figur 3.13 ligger den ikke præcist på 2V. Som under målingen for 10 mmHg skal transducere være i samme højde som målepunktet. Her er målepunktet lavt, men det skaber stadig en del usikkerhed.

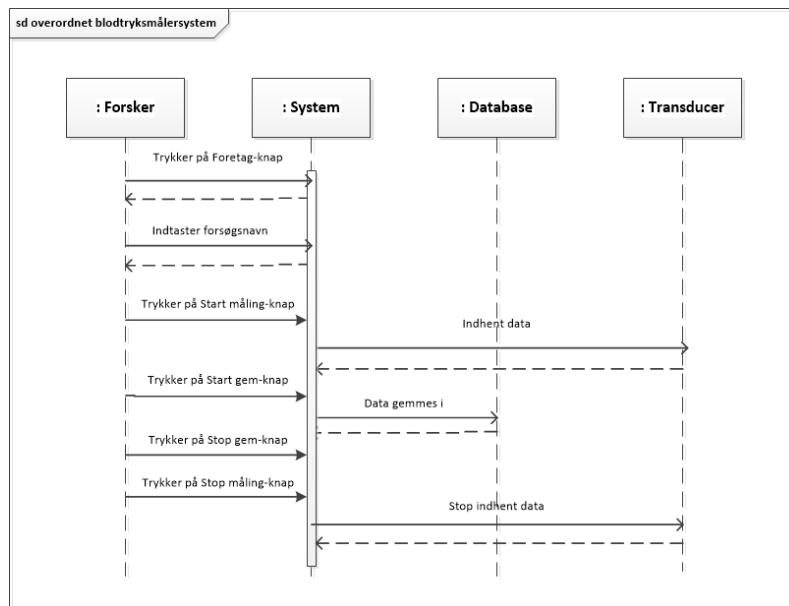
## 3.2 Software

### 3.2.1 Design

I dette beskrives systemets softwaredesign på baggrund af systembeskrivelsen og kravspecifikationen. De overvejelser som er gjort i forbindelse med design af software vil blive præsenteret i dette afsnit.

#### Overordnet sekvensdiagram

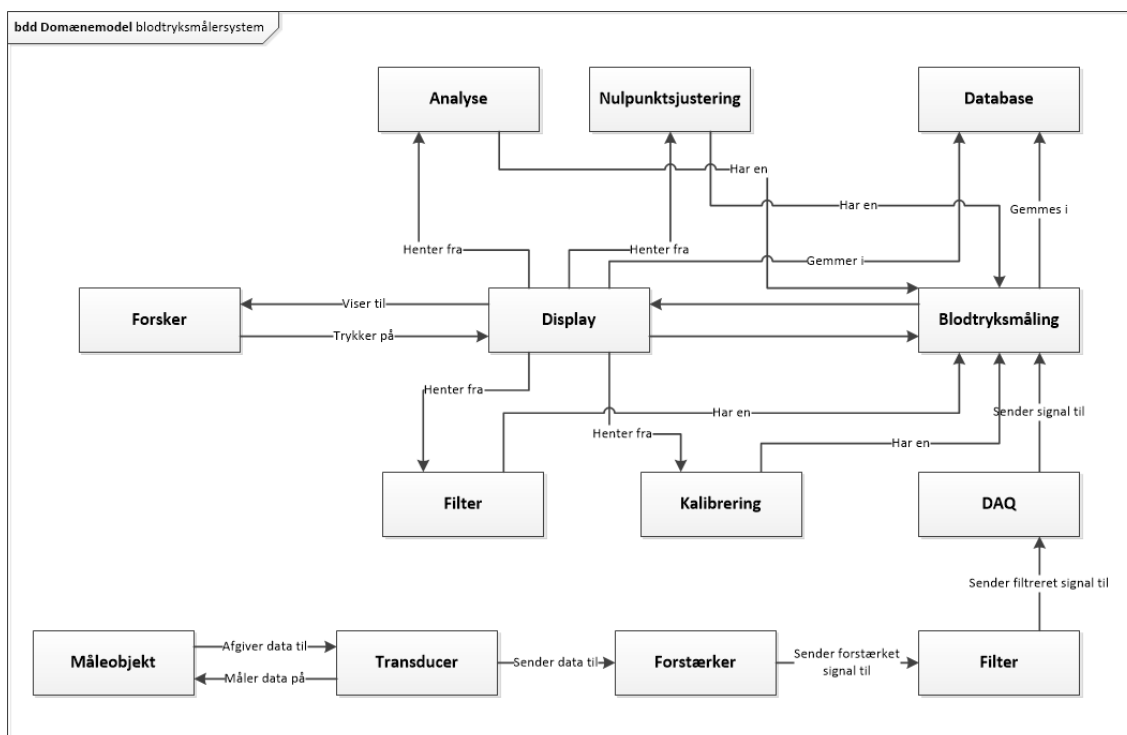
Overordnet set ønskes det at udvikle et system, der kan interagere med en forsker. Diagrammet herunder viser at forskerens opgave består i at starte, tage stilling til nulpunktsjustering og kalibrering samt gemme de ønskede data. Diagrammet er en simpel illustration som viser systemets adfærd gennem alle fem Use Cases. Formålet med dette diagram er udelukkende at skabe et overblik over det samlede system.



Figur 3.14: Overordnet sekvensdiagram for systemet

## Problemidentifikation

Første step i software designet er at klarlægge hvilke klasser systemet skal bestå af. Til dette er en domænemodel derfor udarbejdet med udgangspunkt i de fem Use Cases. I de fem Use Cases er de konceptuelle klasser blevet identificeret, og derefter indført som klasser i nedestående domænemodel. Modellen har til formål at vise hvilke dele systemet skal holde styr på.

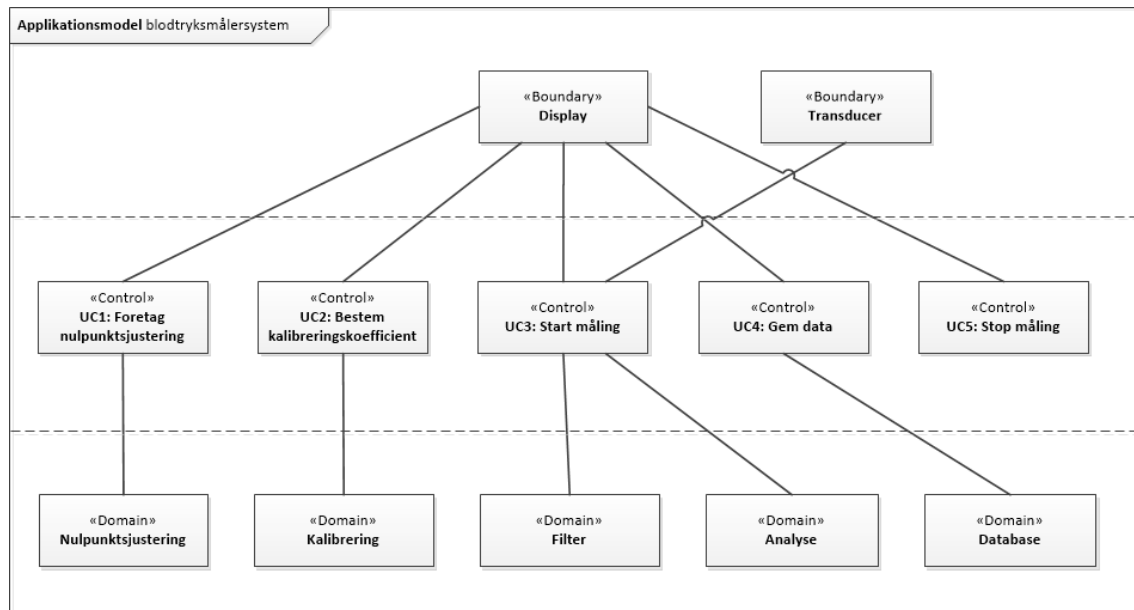


Figur 3.15: Domænemodel

Diagrammet viser tydeligt forskerens interaktion med display, samt hvilke handlinger denne interaktion starter i system. Hardware-komponenterne er medtaget for at vise signalets vej fra måleobjekt til system.

### Klasseidentifikation

Ud fra domænemodellen kan et klassediagram udarbejdes, således tager dette diagram også udgangspunkt i de fem Use Cases. Hensigten med et klassediagram er at klarlægge hver classes individuelle formål.



Figur 3.16: Applikationsmodel for software

Dermed ses det at denne model er delt op i tre niveauer:

#### 1. Grænsefladeklasse

- a) Transducer - Indhentet data fra måleobjekt
- b) Display - Brugergrænseflade til forsker

#### 2. Kontrolklasse

- a) UC1: Foretag nulpunktsjustering
- b) UC2: Bestem kalibreringskoefficient
- c) UC3: Start måling
- d) UC4: Gem data
- e) UC5: Stop måling

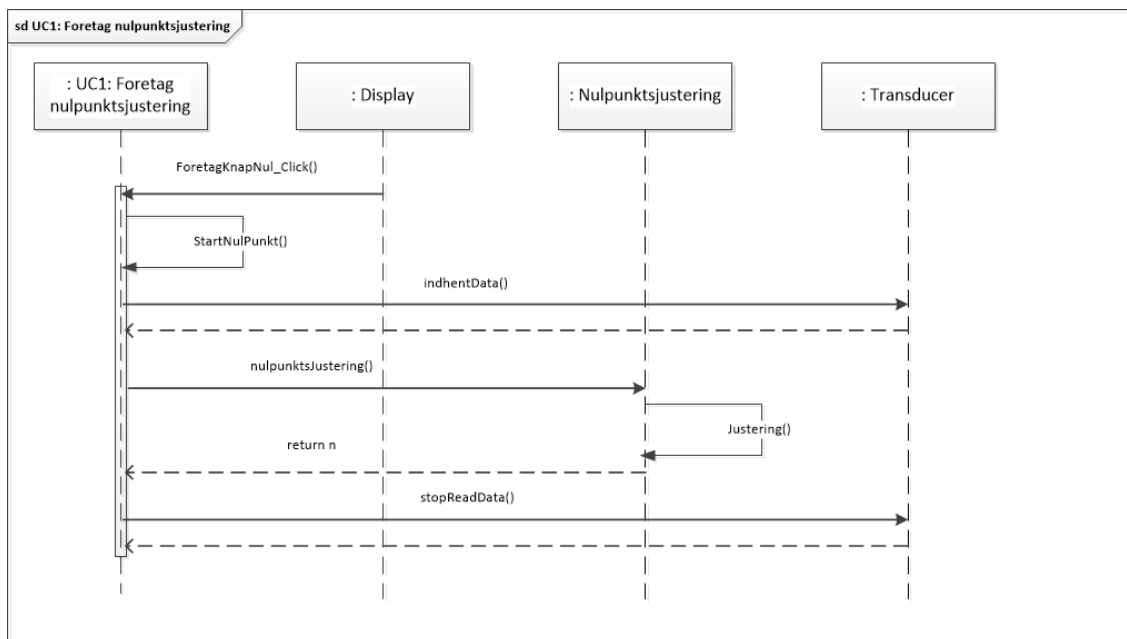
#### 3. Domæneklasse

- a) Database

## Metodeidentifikation

Klasserne i ovenstående klassediagram er med til at definere, hvilke blokke de følgende sekvensdiagrammer må indeholde. Det er yderst vigtigt at der er en sammenhæng mellem klasserne i klassediagrammet og blokkene i sekvensdiagrammet. Vi har valgt at udarbejde et sekvensdiagram for hver enkelt Use Case, hvori systemets interne kommunikation beskrives, når normalforløb og udvidelser gennemløbes. I alle diagrammerne beskrives forløbet via de metodekald, der er nødvendige for at få de ønskede handlinger mellem blokkene udført.

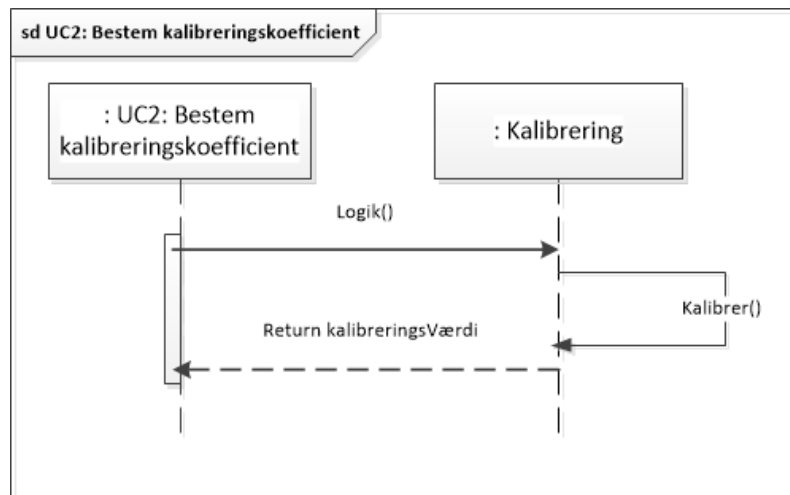
### Use Case 1



Figur 3.17: Sekvensdiagram for Use Case 1

Det ses af ovenstående sekvensdiagram at forsker interagerer med display ved tryk på en knap. Denne interaktion skal igangsætte en nulpunktsjustering, som systemet udfører ved at læse den første indhentede værdi fra transduceren.

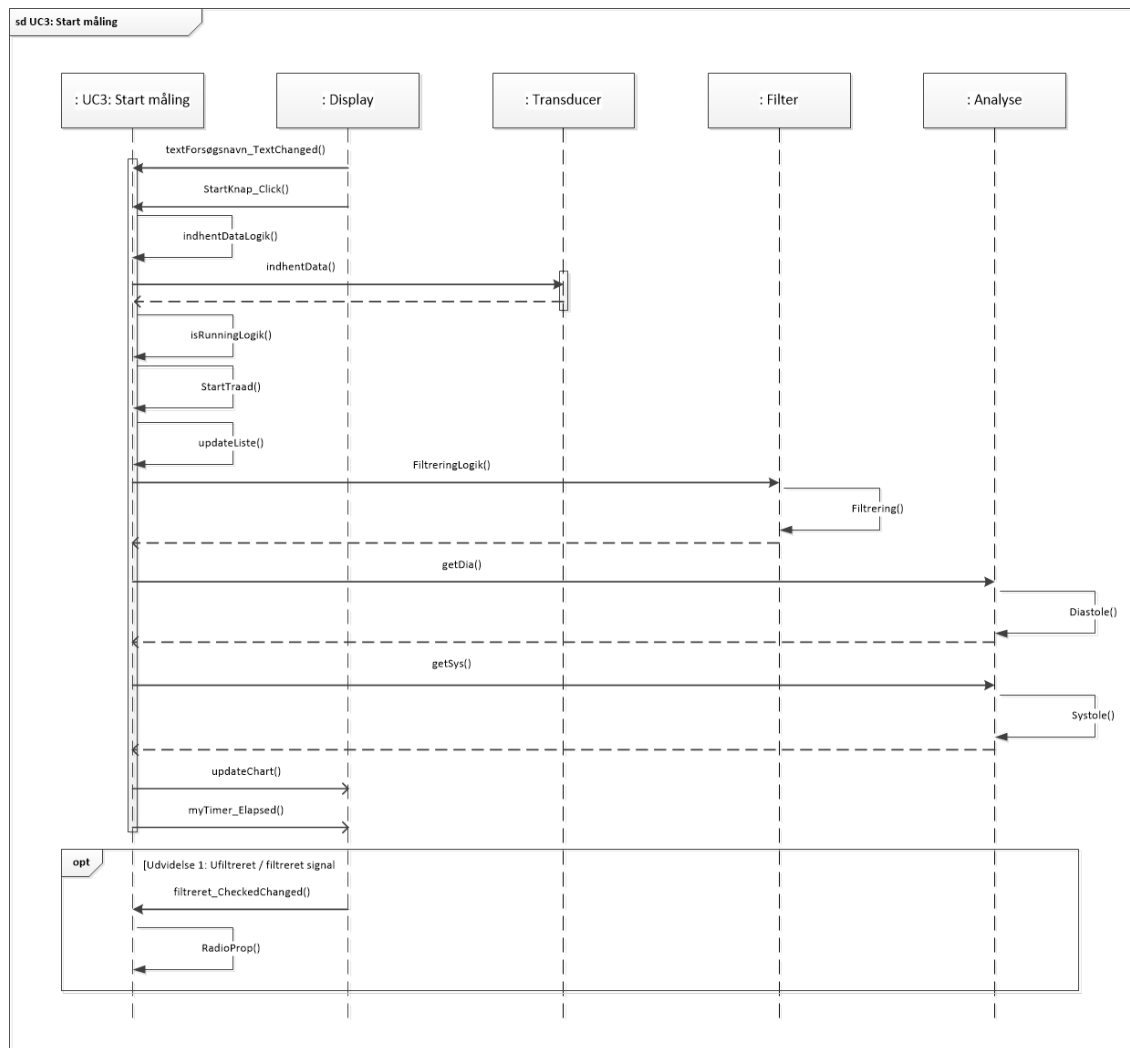
### Use Case 2



Figur 3.18: Sekvensdiagram for Use Case 2

Af diagrammet på figur 3.18 ses det at kalibreringen skal implementeres simpelt i softwaren, hvor kalibreringskoefficienten hentes frem så den kan ganges på samtlige indhentedede samples.

### Use Case 3

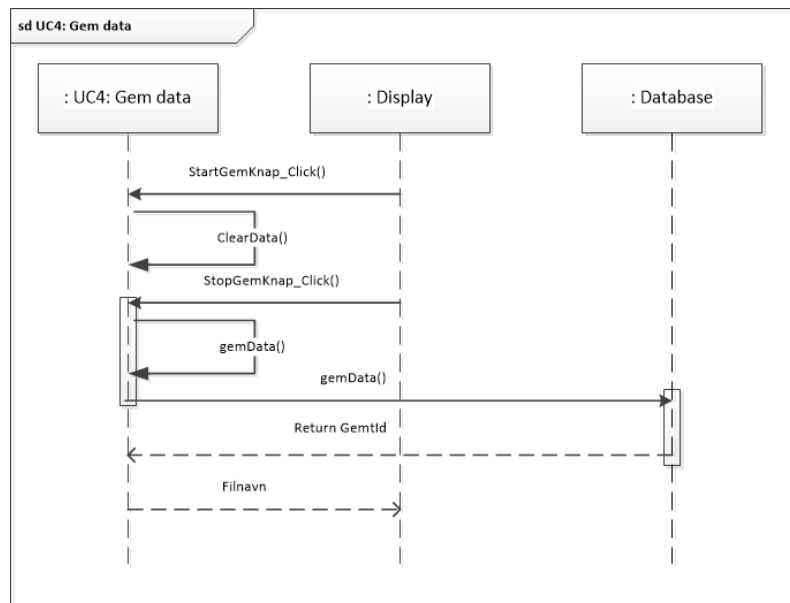


Figur 3.19: Sekvensdiagram for Use Case 3

Ved Use Case 3 - Start måling ses det at display-, transducer-, analyse- og filterklassen vil komme i spil. Her modtages besked ved indtastning af forsøgsnavn og tryk på start-knap på display om, at signaldata fra transducere skal hentes ind i systemet. Herefter foretages filtrering af signalet, samt visning af signal i graf, systoliske-, diastoliske og puls-værdier på display. Use Casen indeholder en udvidelse hvor filtrering af signal ikke ønskes foretaget, dette er vist ved en optional nederst i diagrammet.

#### Use Case 4

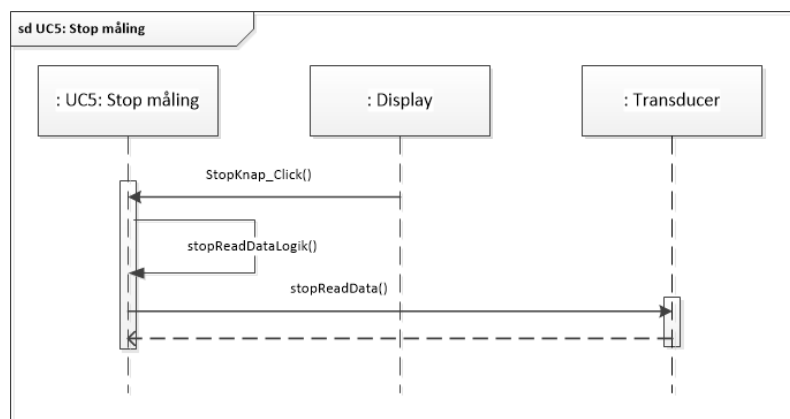




Figur 3.20: Sekvensdiagram for Use Case 4

Ovenstående diagram viser at for at få gemt data fra signalet, kræver det at der trykkes på START GEM-knap på display, hvorefter systemet skal gemme det fremadrettede signal indtil der trykkes på STOP GEM.

### Use Case 5



Figur 3.21: Sekvensdiagram for Use Case 5

Ved stop af en måling ses det at forsker trykker på STOP MÅLING-knap på display, hvorefter indhentning af data fra transducere stoppes.

### 3.2.2 Implementering

#### Indledende implementeringsovervejelser

På baggrund af designfasen for softwaren kan implementeringen af softwaren påbegyndes. Softwaredesignet viser at systemet skal implementeres med en GUI applikation, som aktøren kan interagere med systemet gennem. Derudover er det kendt at softwaren skal

indeholde en række klasser, hvor i funktionaliteter som kalibrering, nulpunktsjustering, digitalt filter og indhentning af systolisk-, diastoliske- og puls-værdier skal placeres. I det følgende beskrives de overvejelser vi har gjort i forhold til implementering af disse funktionaliteter og hele softwaresystemet generelt.

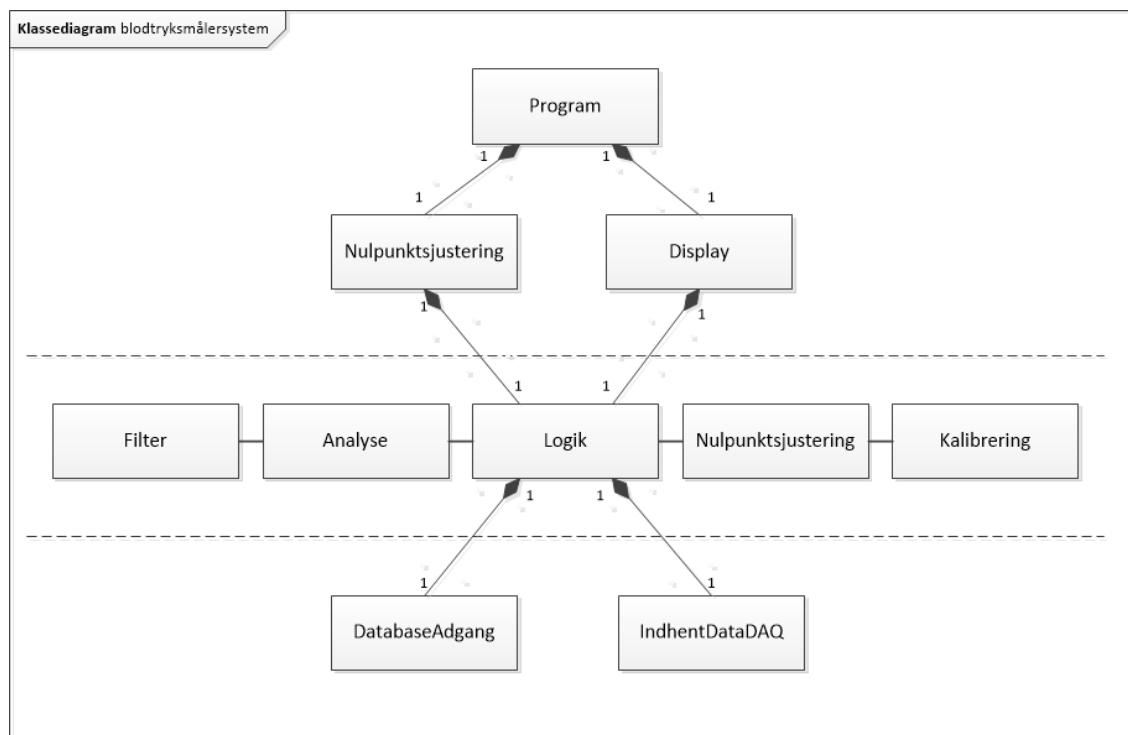
Implementeringen af softwaren sker i Visual Studio 2013 i sproget C#. Dette er valgt da programmet er godt til arbejde med GUI applikationer, samt til håndtering af tråde og tråd kommunikation. Tråde benyttes i softwaren, da systemet der skal implementeres er et eventdrevet system, hvilket vil sige at systemet skal kunne håndtere mange handlinger på en gang. Handlingerne igangsættes af events der kommer af aktørens interaktion med systemet. Tråd kommunikationen fungerer således at en tråd kan sende et signal ud som andre tråde kan reagere på.

Det er valgt kun at udarbejde aktivitetsdiagrammer for metoder, hvor vi mener det er med til at skabe et bedre overblik. Flere at metoderne er simple og derfor er et aktivitetsdiagram ikke nødvendig.

### **Klasse implementering**

På baggrund af designmodellerne er det besluttet at opbygge systemkoden efter principperne i en trelagsmodel. Trelagsmodellen indeholder et præsentrations-lag, et logik-lag og et data-lag. Præsentrations-laget består af de klasser som systemets aktører har tilgang til. Logik-laget er det analyserende lag. Det er således i dette lag at signalet behandles. Logik-laget har tilgang til de andre lag som det eneste. Det betyder at præsentrations-laget og data-laget ikke kan kommunikere sammen, derved skal denne kommunikation foregå gennem logik-laget. Data-laget er tilgangen til den implementerede database og til indhentning af blodtrykssignalet fra hardware.

Fordelen ved trelagsmodel opbygningen er at det skaber et godt overblik i koden, og skaber en kode med lav kobling, da hver enkelt klasse har hvert sit specifikke ansvar. Hvilket gør at koden er let at vedligeholde og ændre hvis funktionaliteter ønskes opbygget anderledes. Et overordnet klassediagram over systemet er udarbejdet på baggrund af præcisering af applikationsmodellen, se figur 3.22. Hvilke metoder hver enkelt klasse indeholder kan ses på de følgende sider, hvor klassediagrammet er blevet brudt op i hver klasse for sig.

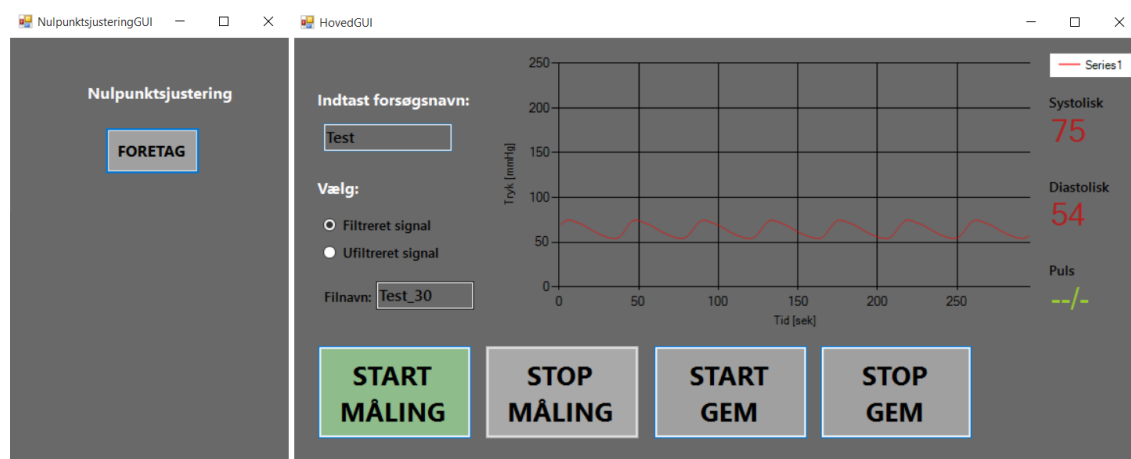


Figur 3.22: Klassediagram

### Brugergrænseflade

Displayet (GUI) er aktørens, i dette tilfælde forskerens, indgang til systemet. Derfor er det vigtigt at den er opbygget efter hvad der følger forskerens logik. Til at klarlægge dette er principperne om en god brugergrænseflade taget i mente. Brugen af disse kommer til udtryk ved, at det tydeligt fremgår af hver knap eller label hvad dens formål er, samt at størrelsen af det enkelte komponent er tilstrækkelig stor til at det ikke er til at overse. Komponenterne på display er logisk placeret, det vil sige at de dele som forsker først skal forholde sig til og eventuelt udfylde er placeret i venstre side af display. Dette vil give mening såfremt systemet benyttes af personer fra den vestlige verden, hvor læseretningen er fra venstre mod højre.

Det er et krav at forsker indtaster et forsøgsnavn inden at en måling startes, derfor er komponenterne implementeres således at knappen "Start måling" først bliver aktiveret når der er indtastet noget i tekstboksen hvori forsøgsnavn skal indtastes. Systoliske-, diastoliske og puls-værdi er placeret efter hvilken rækkefølge det typisk ses på standard blodtryksapparater.



Figur 3.23: NulpunktsjusteringGUI og HovedGUI

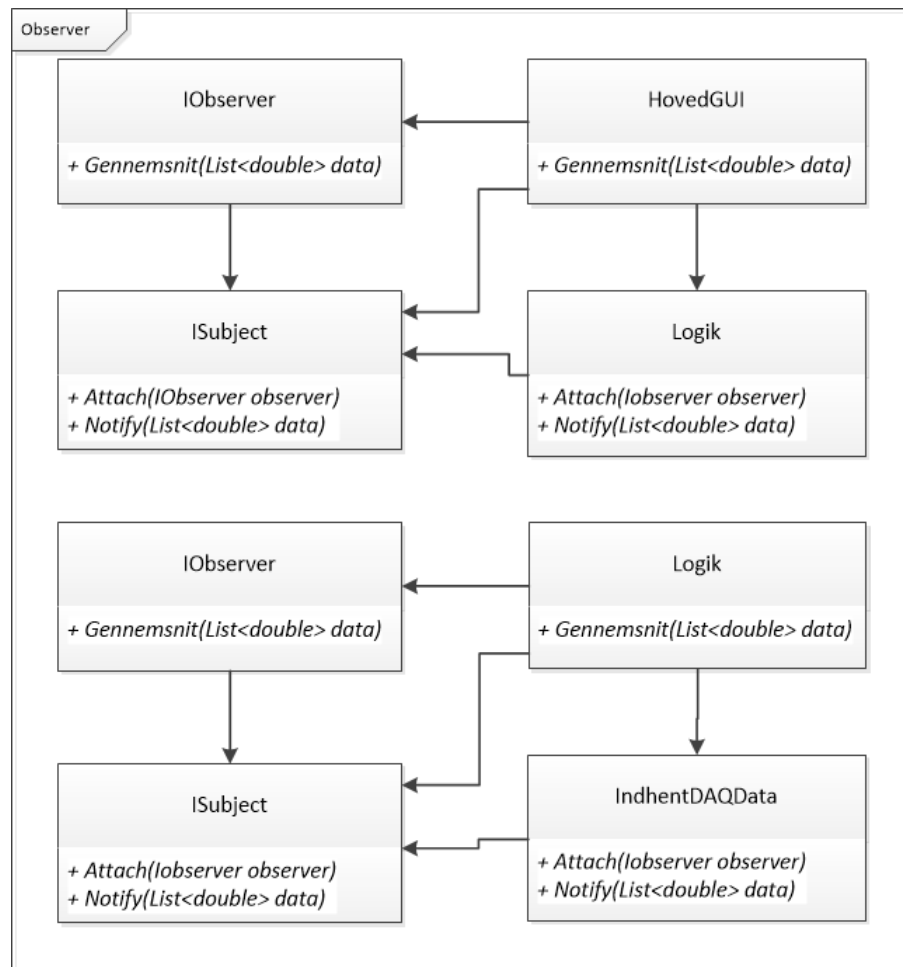
Af figur 3.23 ses det at grafen er en væsentlig del af display's brugergrænseflade. Grafen implementeres som en Windows Form komponent. Det vælges at få vist signalet som en kurve, og førsteaksen indstilles til tid i sekunder fra 0 til 7 sekunder, og andenaksen til en minimums værdi på 0 mmHg og en maksimum værdi på 250 mmHg, hvilket er givet i kravspecifikationen.

### Observer - Strategy

Observer og strategy er to programmeringsmønstre. Der i samarbejde med hinanden er gode til at håndtere at sende data fra et lag til et andet lag. Det er valgt at bygge softwarekoden op efter disse to mønstre. Observer definerer et en til mange forhold mellem objekter således at en ændring i et objekts tilstand medfører at de mange objekter informeres om ændringer og dermed opdateres automatisk.

Dette implementeres ved at oprette to interfaces IObserver og ISubject. Disse interfaces placeres i deres eget namespace, som alle lag kan tilgå, samt gør det muligt for alle nødvendige klasser arve fra disse interfaces. I ISubject placeres de generelle metoder Notify() og Attach(), hvis ansvar er at informere og flytte data fra en klasse til en anden klasse når de kaldes i Subject-klassen. IObserver indeholder metoden der kaldes i Observer når en Notify() fra Subject og ISubject modtages.

Mønstret opbygges som en push, hvilket vil sige at når Subject har ny data klar til at sende op til Observer, kaldes metoden Notify() indeholdende data'en som parametre og dette sendes op til Observer, via ISubject og IObserver. Således fortsætter koden med at arbejde så længe ny data ønskes flyttes op. Mønstret benyttes både mellem data-laget og logik-laget, og mellem logik-laget og præsentations-laget. Skematisk er det i dette projekt givet ved, hvor de relevante metoder i forhold til mønstret er medtaget, se figur 3.23.



Figur 3.24: Observer mønstre

Strategy mønstret indkapsler algoritmer og gør dem udskiftelige med hinanden. Det vil sige at en metode oprettes i et interface. Klasser vil så arve fra dette interface, afhængig af hvem der bruger metoden vil metoden så blive overskrevet i klassen og den nødvendige funktion tilføjet. I samarbejde med Observer-mønstret bruges det ved at Subject arver fra ISubject, og Observer arver fra IObserver. I projektet blev mønstrene i første omgang først benyttet fra logik-laget til præsentations-laget i forbindelse med at sende data til visning i graf. Men undervejs viste det sig nødvendigt også at implementere mønstrene fra data-laget til logik-laget, således at det kan kontrolleres hvor stor en mængde data der sendes op ad gangen.

### Samplefrekvens

Samplefrekvensen er som krav givet til 1000 Hertz. Hvilket svarer til at systemet modtager 1000 samples i sekunder. Varigheden af en sample er givet ved:

$$\frac{1}{f_s} = \frac{1}{1000} = 0.001 \text{ sek} \quad (3.14)$$

Det har vist sig under arbejdet med softwaren, at systemet ikke kan følge med til at modtage så mange målinger i sekundet. Derfor er det valgt at skære i antallet af målinger pr. sekund der skal videre bearbejdes i logik-laget og udskrives i præsentations-laget. Antallet

skæres ned til 50 målinger pr. sekund. Dette gøres ved at gennemsnittet af 20 målinger efter hinanden bestemmes, hvorefter gennemsnitsværdien returneres og gemmes i listen der sendes videre i systemet. Herefter findes så gennemsnittet af de næste 20 målinger og således fortsættende.

### Nulpunktsjustering

Formålet med en nulpunktsjustering er at flytte signalets offset enten op eller end, så det atmosfæriske tryk altid er placeret ved 0 volt på outputsignalet. Dette gøres ved at åbne for den tilsluttede transducer til systemet, så det atmosfæriske tryk måles. Ud fra denne værdi kan justeringsfaktoren så bestemmes ved, hvor  $x$  er det målte atmosfæriske tryk i volt modtaget gennem DAQ'en:

$$faktor_{jus} = 0 - (x) \quad (3.15)$$

Af ligningen ses det at justeringsfaktoren både vil kunne blive positiv og negativ, afhængig af om offset værdien skal rykkes op eller ned for at blive placeret i nul. Optimalt set vil det atmosfæriske tryk være en konstant værdi ved den samme måling, men det opleves at der er en smule støj på signalet og derfor vil den målte værdi være en tilnærmelse af det atmosfæriske tryk. Systemet ønskes nulpunktsjusteret for at sikre at alle de målte blodtrykssignaler har samme udgangspunkt. Hvilket gør at målingerne kan sammenlignes. Systemet foretager automatisk nulpunktsjusteringen når systemet startes ved at retunerer den første værdi fra DAQ'en, når der trykkes på knappen FORETAGET. Denne værdi er justeringsfaktoren der lægges til samtlige samples i det indhentede blodtrykssignal.

### Kalibrering

Ved kalibrering ønskes det at bestemme hardwarens visningsfejl. I dette projekt betyder det at kalibreringsfaktoren fra volt til millimeter kviksølv bestemmes. Denne bestemmes ved at tilkoble en væskesøjle til systemet. Væskesøjlen fyldes med vand til den vil give et kendt mængde tryk på systemet angivet i mmHg. Herefter kan output i volt fra hardwaren måles. kalibreringsfaktor er givet ved:

$$faktor = \frac{x[mmHg]}{y[Volt]} \quad (3.16)$$

$x$  angiver trykket fra væskesøjlen, denne hardcodes til 50 mmHg.  $y$  angiver den målte spændingsoutput på hardwaren. Optimalt set er kalibreringsfaktoren givet ved:

$$\frac{250[mmHg]}{5[V]} = 50 \quad (3.17)$$

hvor 250 mmHg er det maksimale blodtryk systemet kan måle og 5 Volt er maks spændingen i volt. Grafisk vil det se ud som vist på figur 3.8 under hardware modultest. Af figur 3.8 kan det aflæses at den optimale outputspænding ved 50 mmHg er 1 Volt. Kalibreringsfaktoren skal ganges på samtlige sample-værdier der kommer fra DAQ'en og som ønskes udskrevet på graf i display. Kalibreringen implementeres i softwaren ved brug af konfiguration. Forskeren beregner omsætningsværdien ud fra ligning 3.16. Resultatet af denne beregning indtaster forsker i konfigurations xml-filen under App.settings. XML-filen kan tilgås uden opstart af systemet, derfor bliver kalibreringen uafhængig af hvornår

systemet kører og kalibreringen kan dermed foretages på et vilkårligt tidspunkt. Værdien der ændres i XML-filen er den tilhørende "Value" til "KalibreringsKoefficient". Den er markeret med grøn firkant omkring på figur 3.24.

```

1  <?xml version="1.0" encoding="utf-8" ?>|
2  <configuration>
3  <startup>
4      <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
5  </startup>
6  <appSettings>
7      <add key="KalibreringsKoefficient" value="50"/>
8  </appSettings>
9  </configuration>

```

Figur 3.25: Konfigurations XML-fil

Metoden Kalibrering() i Kalibreringsklassen, som er en del af logik-laget læser så "Kalibreringskoefficienten" fra konfigurations-filen hver gang kalibreringsfaktoren skal ganges på et signal.

Det er vigtigt at pointere at nulpunktsjusteringsfaktoren lægges til samtlige værdier i signalet førend at kalibreringsfaktoren ganges på. Dette udføres i kodens logik-lag.

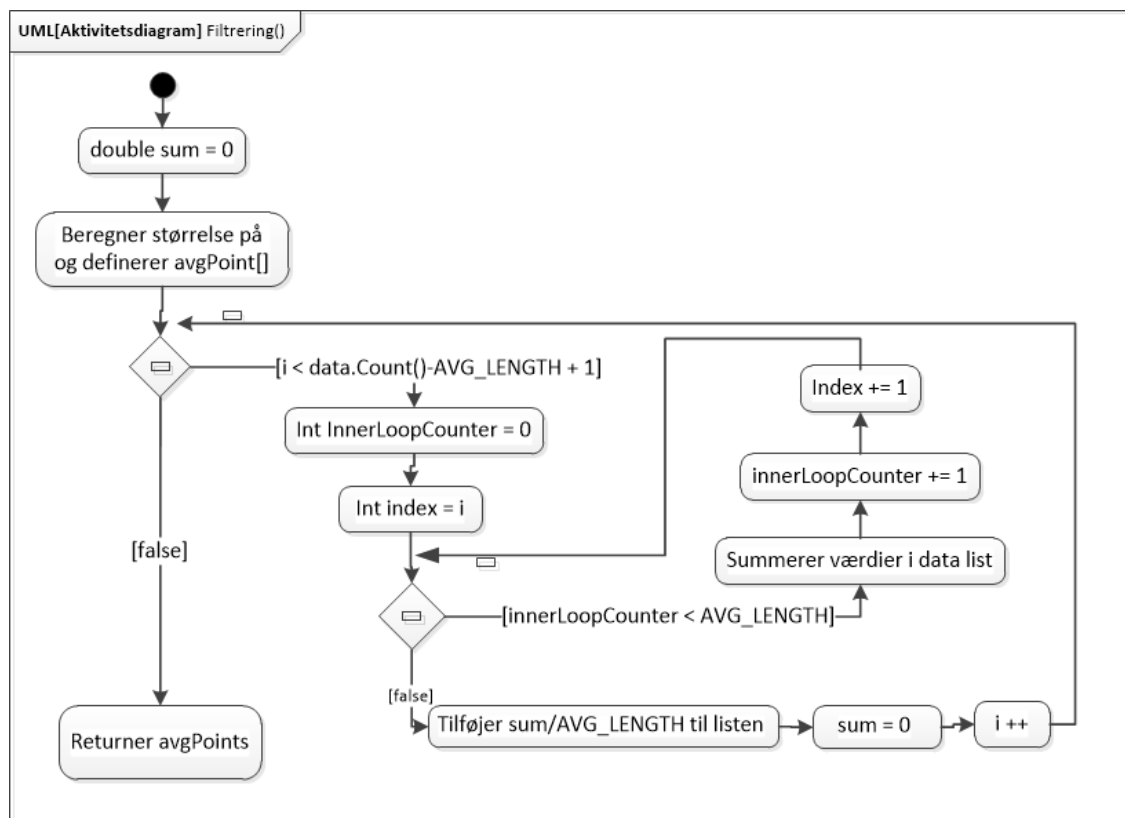
### Digitalt Filter

[?] Formålet med implementering af et digitalt filter er at fjerne støj fra det indhentede signal. Dette gøres ved at udglatte signalet. Til dette kan en række forskellige filtre benyttes. Vi har valgt at implementere et glidende middelværdifilter (moving average filter). Fordelen ved dette filter er at det er simpelt at forstå og at det er optimalt at bruge på signaler i tidsdomænet. Skulle signalet være vist i frekvensdomænet ville valget have faldet på et helt andet filter.

Det glidende middelværdifilter fungerer ved midling af en række punkter fra inputsignalet for at frembringe hvert punkt i outputsignalet. Hvilke punkter der tages fra inputsignalet vil flytte sig en plads for hvert beregnet outputsignal punkt, heraf kommer den glidende effekt. Matematisk er filtret givet ved:

$$y[i] = \frac{1}{M} \cdot \sum_{j=0}^{M-1} x[i+j] \quad (3.18)$$

Hvor  $x[]$  er inputsignalet,  $y[]$  er outputsignalet og  $M$  er antallet af punkter der benyttes i det glidende middelværdifilter. Denne beregning benytter sig udelukkende af punkter placeres på den samme side af output sample nummeret, hvilket vil føre til en relativ forskydning mellem input og output.  $M$  sættes til 5. Implementeringen af filtret er vist i et aktivitetsdiagram på figur 3.25.



Figur 3.26: Aktivitetsdiagram af metoden Filtrering()

Måden hvorpå filtret er implementeret gør at der ikke sker en filtrering af de første fire samples, det ses af følgende i koden. AVG\_LENGTH er defineret til 5, og mængden af punkter der benyttes i filtret i ligning 3.18 svarer dette til M.

```

1 reference
public List<double> Filtrering(List<double> data)
{
    double sum = 0;
    List<double> avgPoints = new List<double>();
    for (int i = 0; i < data.Count() - AVG_LENGTH + 1; i++)
    {

```

Figur 3.27: Udsnit af koden til det glidende middelværdifilter

Det ses at der skal være minimum 5 samples i data.Count førend at listen avgPoints oprettes. Det er en begrænsning vi er opmærksom på, men som accepteres da de første fire samples ved visning i graf er kørt så hurtigt igennem, at det ikke skaber en begrænsning for brugen af systemet for forsker. Optimalt set vil der sættes en begrænsning på filtret således når første måling modtages vil gennemsnittet findes af en sample, dernæst af to samples, tre samples osv. Indtil der er fem samples og gennemsnittet vil så altid bestemmes af de fem seneste samples.

Systemet gør det muligt for forsker selv at vælge om signalet ønskes vist filtreret eller ufiltreret. Dette vælges på brugergrænsefladen. Vælges visning af det ufiltrerede signal sendes det indhentede signal naturligvis ikke gennem det digitale filter. Det er muligt at skifte



mellem filtreret og ufiltreret signal, mens systemet kører. I det tilfælde skifter hele det viste signal til det valgte, da alt data i listen der indhentes dermed skifter. Filtreringen vil dermed ikke vise sig som en løbende kurve grafisk.

### Analyse

Analyse dækker over indhentningen af de systoliske-, diastoliske- og puls-værdi ud fra blodtrykssignalet. Dette er implementeret i en klasse kaldet Analyse. Heri er placeret metoder for henholdsvis systole og diastole. I en blodtrykskurve er den systoliske værdi givet ved maximum på kurven og den diastoliske er givet ved minimums værdien på kurven. Metoderne bestemmer derfor den maksimale værdi og den mindste værdi i listen, der medtages som parametre til metoderne. Listen der bruges som parametre er UILIST indeholdende 350 tal ad gangen. UILIST er listen der sendes fra logik-laget til præsentations-laget med de behandlede data, som vises i grafen. I præsentations-laget er implementeret en timer, der håndterer at de systoliske- og diastoliske værdier i display kun opdateres hvert 3 sekund. I løbet af 3 sekunder vil der være gennemløbet 3-5 blodtryksperioder, afhængig af pulsfrekvensen. Dermed vil samtlige systoliske og diastoliske værdier ikke blive udskrevet. Intervallet på 3 sekunder er valgt da det er passende tid til at kunne nå og aflæse den pågældende værdi.

I forhold til implementering af puls er der gjort en række overvejelser og mulige løsninger. Puls er defineret ved slag pr. minut og på en puls vil der være en systole og diastole. Pulsen må derfor kunne bestemmes ved at tælle antallet af systoliske værdier på 6 sekunder, antallet ganges så med 10 for at få den rette enhed. Udfordringer er dog opstået i forhold til at kunne bestemme præcist hvornår der er gået 6 sekunder i programmet. En anden mulighed er også at bestemme pulsen ved at finde antallet af samples mellem to systoliske værdier. Omregnes samples så til sekunder og ganges op til et minut, må dette være ligmed måleobjektets øjeblikkelige puls. Det er dog ikke lykkedes at omsætte overvejelserne til kode, og pulsen er dermed ikke blevet implementeret ved projekt aflevering.

### Beregnings metode

Metoden `updateListe()` er den vigtigste metode i logik-laget. Den er placeret i logikklassen. Metoden har til ansvar at tage højde for nulpunktsjustering og kalibrering ved beregning. Derudover har den til ansvar at lægge data i en liste med 300 pladser, der kan vises i graf på display. Til sidst tjekker metoden om radiobutton for filtreret eller ufiltreret signal er checked og sender listen gennem filtret, hvis filtret signal er valgt inden listen med `Notify()` sendes til præsentationslaget. Metodens aktivitetsdiagram er givet ved:

### Database

I systemet er der implementeret en lokal database. Databasen er oprettet gennem host `webhotel10.iha.dk`. Formålet med databasen er at lagre det målte blodtrykssignals rådata. Det er valgt at implementere databasen som typen SQL, da denne database-type indeholder de funktioner som er nødvendige for dette system. Data gemmes i denne type database i tabeller. Indledningsvis for at oprette den nødvendige tabel defineres en type til hver værdi. SQL-koden til oprettelse af tabel er vist på figur xx.

```
1 CREATE TABLE [db_owner].[SEMPRJ3] (  
2     [Forsøgsnavn] NVARCHAR (20) NOT NULL,  
3     [Id] BIGINT IDENTITY (1, 1) NOT NULL,  
4     [Datostempel] DATETIME NOT NULL,  
5     [Blodtryksmåling] VARBINARY (MAX) NOT NULL,  
6     PRIMARY KEY CLUSTERED ([Id] ASC)  
7 );
```

Figur 3.28: SQL-kode til oprettelse af tabeller i database

Forsøgsnavnen referer til det forsøgsnavn der indtastes i GUI ved påbegyndelse af en ny måling. Dette er af typen NVARCHAR(20), hvilket betyder at forsøgsnavnet maksimalt kan være 20 tegn langt. Id er defineret som primær nøgle, det betyder at denne er unik for hver enkelt sekvens i database, og Id der vil referes til mellem tabeller i databasen, hvis flere tabeller var nødvendigt.

Et blodtrykssignal indeholder en stor mængde datapunkter, derfor gemmes signalet i en VARBINARY, hvor en række binære datapunkter gemmes som en enkelt enhed i databasen. Dette er valgt for at spare på data pladsen i databasen. Denne type besværliggør dog, at få vist hvilke værdier blodtrykssignalets datapunkter består af.

Databasen er implementeret således at flere sekvens af den samme måling kan gemmes uden at systemet skal startes forfra. Dette er smart for forsker, hvis der testes flere ting på det samme signal.

### 3.2.3 Modultest

# Accepttest 4

---

Version	Dato	Ansvarlig	Beskrivelse
0.1	28-09-2015	MHNC og MBA	Oprettelse og udfyldelse af Accepttest
0.2	30-09-2015	ABH	Tilrette accepttest
0.3	08-10-2015	Alle	Tilrette efter review med Grp. 1
0.4	15-10-2015	MBA	Indskrevet i LaTeX
0.5	20-10-2015	MHNC	Tilretning
0.6	26-11-2015	MHNC	Retning af hele accepttesten. Konsekvent med stavemåder
0.7	10-12-2015	DHC, ABH, AJF	Rettelser i forhold til slutprodukt

## 4.1 Accepttest af Use Cases

## 4.2 Indledning

Accepttestene skal vise om produktet lever op til de standarder vi har sat op for, at den aktivt kan indgå i en forskningssituation. Accepttesten er en opfølgning af kravspecifikation, som har til formål at sikre at alle kravene er overholdt. Der vil blive testet både på hovedscenarier samt på undtagelser. Det er målsætningen, at disse test sikrer produktets kvalitet, idet produktet vil blive afprøvet før det tages i brug. Derfor er det accepttestens ansvarsfunktion, at godkende de opsatte delmål for produktet hvad angår både funktionalitet samt ikke-funktionelle krav.

Data der benyttes til målingerne fås fra In Vitro, der i form af tryk genererer et fysiologisk tryk. Brugergrænsefladen er det som forskeren initierer med, altså hvorfra systemet aktiveres. Brugergrænsefladen forkortes til GUI. Den benyttede Database er en lokal database. Når der i feltet Godkendt er et flueben, betyder det at testen er godkendt. Hvis der er et flueben i parenteser, betyder det at den er delvis godkendt.

### 4.2.1 Use Case 1

Indsæt beskrivelse og figurer med NI-DAQ, Analog discovery og transduceren. Det forventes for Use Case 1, at forskeren har fået påmonteret det væskefyldte kateter samt tændt for apparaturet.

Test af Use Case 1	Foretag nulpunktsjustering
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren og pop-up vindue for nulpunktsjustering er åbent

Handling	Forventet observation/resultat	Faktisk observation/resultat	Godkendt
<i>Hovedscenarie</i>			
1. Forsker trykker på Foretag-knap	Systemet foretager nulpunktsjustering, hvorefter vinduet lukker		

Tabel 4.3: Accepttest af Use Case 1

#### 4.2.2 Use Case 2

Test af Use Case 2	Bestem kalibreringskoefficient
Scenarie	Hovedscenarie
Prækondition	Hardware er monteret ved 50 mmHg på væskesøjlen og er tilkoblet en computer med WaveForm.

Handling	Forventet observation/resultat	Faktisk observation/resultat	Godkendt
<i>Hovedscenarie</i>			
1. Output spænding fra hardware aflæses i WaveForm	Output aflæses til 1 V +/- 30%		
2. Beregning foretages ud fra formelen $\frac{50}{output} = koefficient$	Koefficienten beregnes til 50 +/- 30%		

3.	Forsker indtaster beregnet kalibreringskoefficient i konfigurations XML-fil	Koefficienten står i XML-fil
4.	Kalibreringskoefficient kan tilgås af systemet	Sammenlign værdierne i listen råData(findest i IndhentDAQData) med den viste graf på GUI

Tabel 4.5: Accepttest af Use Case 2

### 4.2.3 Use Case 3

Test af Use Case 3	Start måling
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren. UC1 er kørt succesfuldt

	Handling	Forventet observation/resultat	Faktisk observation/resultat	Godkendt
<i>Hovedscenarie</i>				
1.	Forsker indtaster Forsøgsnavn.	Systemet tilgængeliggør Start Måling-knap		
2.	Filteret signal er valgt per default af systemet	Radiobutton til filtret signal er checket af		
3.	Forsker trykker på Start Måling-knap på GUI	Signal vises i graf på GUI		
4.	Systolisk og diastolisk blodtryk samt puls bliver vist i bokse på GUI	GUI udskriver systoliske, diastoliske og puls værdier på GUI		

*Udvidelse 1: Forsker  
vælger filtreret/  
ufiltreret signal*

---

1.	Forsker vælger ufiltreret signal.	Grafen viser det ufiltreret signal
2.	Forsker vælger filtreret signal.	Grafen viser det filtreret signal

---

*Tabel 4.7: Accepttest af Use Case 3*

#### 4.2.4 Use Case 4

---

Test af Use Case 4	Gem data
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren. Use Case 1 er kørt succesfuldt, Use Case 3 kører

---

Handling	Forventet observation/resultat	Faktisk observation/resultat	Godkendt
----------	--------------------------------	------------------------------	----------

---

*Hovedscenarie*

---

1.	Forsker trykker på Start Gem-knap	Start Gem-knap bliver highlightet med blå kant	
2.	Forsker trykker på Stop Gem-knap for at stoppe med at gemme	Filnavnet(forsøgsnavn_Id) bliver vist i tekstboks GUI	
3.	Forsker trykker på Gem-knap for at stoppe med at gemme	Det fremgår af GUI at data er gemt i Database	

---

*Undtagelse 1: Forsker  
trykker på Stop  
Måling-knap*

---

1. Forsker trykker på Stop Måling-knap til et givent tidspunkt	Grafen på GUI fastholdes og datostempel på seneste indlagte data aflæses i databasens tabel. Tiderne sammenlignes.
--	--

---

*Tabel 4.9: Accepttest af Use Case 4*

#### 4.2.5 Use Case 5

---

Test af Use Case 4	Stop måling
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren. Use Case 1 er kørt succesfuldt, Use Case 3 kører

---

Handling	Forventet observation/resultat	Faktisk observation/resultat	Godkendt
<i>Hovedscenarie</i>			
1. Forsker trykker på Stop Måling-knappen	Målingen stoppes og blodtryksgrafnen fastholdes		

---

*Tabel 4.11: Accepttest af Use Case 5*

### 4.3 Accepttest af ikke-funktionelle krav

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt
1.	Blodtryksmåleren skal indeholde en Start Måling-knap til at igangsætte målingerne.	Kør Use Case 1 og 3	Start Måling-knap er på GUI		
2.	Blodtryksmåleren skal indeholde en Stop Måling-knap, hvorfra måling kan stoppes.	Kør Use Case 1 og 3	Stop Måling-knap er på GUI		
3.	Blodtryksmåleren skal indeholde en Start Gem-knap til påbegyndelses af at gemme måling i Database	Kør Use Case 1 og 3	Start Gem-knap er på GUI		
4.	Blodtryksmåleren skal indeholde en Stop Gem-knap til påbegyndelses af at gemme måling i Database	Kør Use Case 1 og 3	Stop Gem-knap er på GUI		



5.	Blodtryksmåleren skal indeholde en tekstboks til forsøgsnavn, hvori forsker indtaster det pågældende forsøgsnavn.	Kør Use Case 1 og 3	Tekstboks til forsøgsnavn er på GUI
6.	Blodtryksmåleren skal indeholde radiobutton til filtreret signal, denne skal være default valget	Kør Use Case 1 og 3	Radiobutton til filtreret signal er på GUI
7.	Blodtryksmåleren skal indeholde radiobutton til ufiltreret signal	Kør Use Case 1 og 3	Radiobutton til ufiltreret signal er på GUI
8.	Blodtryksmåleren skal indeholde tekstbokse til puls, systolisk og diastolisk blodtryk som vises med op til tre cifre	Kør Use Case 1 og 3	Systolisk-boks, diastolisk-boks og puls-boks er på GUI
8.	Blodtryksmåleren skal indeholde en tekstboks som viser filnavn(forsøgsnavn og id) på målingen, efter måling er gemt.	Kør Use Case 1 og 3	Tekstboks til Filnavn er på GUI

9.	GUI'en skal se ud som på figur 2.3 i KS	GUI'en ser ud som figur 2.3 i KS	GUI'en ser ud som figur 2.3 i KS
10.	Forskeren skal kunne starte en default-måling maksimalt 30 sekunder efter systemet er startet	Systemet er åben samtidigt startes et stopur. Efter tryk på Start Måling-knap og målingen er startet stoppes uret	Måling er startet og stopuret viser mindre end 30 sekunder
10.	Det skal maksimalt tage 5 timer at gendanne systemet (MTTR - Mean Time To Restore)		Kan ikke testes på prototypen
11.	Systemet skal have en opetid uden nedbrud på minimum 1 måned (720 timer) (MTBF - Mean Time Between Failure)		Kan ikke testes på prototypen
12.	Systemet skal have en opetid/køretid på: $\frac{MTBF}{MTBF+MTTR} * 100 = 99,31\%$		Kan ikke testes på prototypen

13.	Blodtryksmåleren skal, indenfor 3 sekunder, kunne vise systolisk og diastolisk blodtryk via graf. Dette accepteres med en tolerance på +/- 15 %	Kør Use Case 1 og 3. Der trykkes på Start Måling-knappen samtidig med at et stopur startes. Når måling vises i graf stoppes uret	Stopuret viser mellem 2.55 - 3.45 sekunder
15.	Blodtryksmåleren skal, indenfor 5 sekunder fra der er trykket på Stop Gem-knap, have gemt målingerne i Databasen. Dette accepteres med en tolerance på +/- 15 %.		Kan ikke testes på prototypen
16.	Grafen vises i ét vindue, hvor y-aksen måles i mmHg og x-aksen i tid i sekunder	Kør Use Case 1 og 3	På GUI er y-aksen målt i mmHg og x-aksen i tid pr. sekund

17.	Hver 3. sekund skal værdier for systolisk og diastolisk blodtryk samt puls opdateres. Dette accepteres med en tolerance på +/- 15 %	Kør Use Case 1 og 3. Forsøgsnummer indtastes og der trykkes på Start Måling-knappen samtidig med at et stopur startes. Når værdier i bokse vises stoppes uret	Stopuret viser mellem 5.95 - 8.05 sekunder
18.	Graf for blodtryk skal køre kontinuerligt i GUI efter princippet på figur 2.4	Kør Use Case 1 og 3	Grafen i GUI kører kontinuerligt efter princippet på figur 2.4
19.	Når der trykkes på Stop Gem-knap gemmes signals rådata under det indtastede forsøgsnavn og et autogenereret id. <i>"forsøgsnavn_id"</i>	Kør Use case 1, 3 og 4	Data er blevet gemt i Databasen under filnavnet <i>"forsøgsnavn_id"</i>
20.	Systemet skal kunne måle blodtryksværdier fra 0 til 250 mmHg	Kør Use Case 1 og 3	Det indhentede signals blodtryksværdier er indenfor 0 til 250 mmHg på grafens y-akse

21.	Forskeren skal kunne udskifte batterierne til hardwaren på 2 minutter.	Udskiftning af batterier påbegyndes samtidig med at stopur startes. Når de er udskiftet stoppes uret	Stopuret viser mindre end 2 minutter
22.	Softwaren skal opbygges med lav kobling	Åbn systemets programkode	Koden er opbygget med lav kobling

*Tabel 4.12: Accepttest af Ikke-funktionelle krav*

## 4.4 Godkendelsesformular

---

Godkendes af Peter Johansen

Kunde IHA

Dato for test \_\_\_\_\_

Ved underskrivelse af dette dokument godkendes den kørte accepttest.

\_\_\_\_\_  
Sted

\_\_\_\_\_  
Dato

\_\_\_\_\_  
Kundens underskrift

\_\_\_\_\_  
Leverandørens underskrift

# Figurer

---

2.1	Aktør-kontekstdiagram . . . . .	4
2.2	Use Case-diagram . . . . .	5
2.3	Skitse af GUI . . . . .	10
2.4	Graf for blodtryk . . . . .	11
3.1	Block Definition Diagram for hardware . . . . .	13
3.2	Internal Block Diagram for hardware . . . . .	13
3.3	Bodeplot . . . . .	17
3.4	Diagram over HW . . . . .	18
3.5	Forstærknings blok . . . . .	18
3.6	Måling for 10 Hz . . . . .	19
3.7	Måling for 50 Hz . . . . .	19
3.8	Måling for 60 Hz . . . . .	20
3.9	Graf til kalibrering, fra udregninger . . . . .	20
3.10	Opstilling . . . . .	21
3.11	Måling ved 50 mmHg . . . . .	21
3.12	Måling ved 10mmHg . . . . .	22
3.13	Måling ved 100mmHg . . . . .	23
3.14	Overordnet sekvensdiagram for systemet . . . . .	24
3.15	Domænemodel . . . . .	24
3.16	Applikationsmodel for software . . . . .	25
3.17	Sekvensdiagram for Use Case 1 . . . . .	26
3.18	Sekvensdiagram for Use Case 2 . . . . .	27
3.19	Sekvensdiagram for Use Case 3 . . . . .	27
3.20	Sekvensdiagram for Use Case 4 . . . . .	28
3.21	Sekvensdiagram for Use Case 5 . . . . .	28
3.22	Klassediagram . . . . .	30
3.23	NulpunktsjusteringGUI og HovedGUI . . . . .	31
3.24	Observer mønstre . . . . .	32
3.25	Konfigurations XML-fil . . . . .	34
3.26	Aktivitetsdiagram af metoden Filtrering() . . . . .	35
3.27	Udsnit af koden til det glidende middelværdifilter . . . . .	35
3.28	SQL-kode til oprettelse af tabeller i database . . . . .	36