



AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI  
3. SEMESTERPROJEKT

---

# Dokumentation

---

*Gruppe 3*

Helle Randeris (studienr.)

Rune Rask (studienr.)

Joakim Lindhardt (studienr.)

Finja Ralfs (studienr.)

Lars Holst (studienr.)

Signe S. Vaaben (201310503)

*Vejleder*

Titel

Navn

Universitet

26. november 2015



*Gruppemedlemmer*

_____	_____
CStuderende (studienr.)	Dato

_____	_____
Studerende (studienr.)	Dato

_____	_____
Studerende (studienr.)	Dato

_____	_____
Studerende (studienr.)	Dato

_____	_____
Studerende (studienr.)	Dato

*Vejleder*

_____	_____
Vejledernavn	Dato



# Forkortelser

---

Forkortelse	Forklaring
BT	blodtryk
UC	Use case
BDD	Block Definition Diagram
IBD	Internal Block Definition
SD	Sekvensdiagram
DAQ	Digital to Analog Converter
GUI	Graphical User Interface (brugergrænseflade)
SW	Software
HW	Hardware
KS	Kravspecifikation



# Indholdsfortegnelse

---

<b>Forkortelser</b>	<b>iii</b>
<b>Kapitel 1 Kravspecifikation</b>	<b>1</b>
1.1 Systembeskrivelse . . . . .	1
1.2 Funktionelle krav . . . . .	2
1.2.1 Aktør-kontekstdiagram . . . . .	2
1.2.2 Aktørbeskrivelse . . . . .	3
1.2.3 Use case-diagram . . . . .	3
1.2.4 Use cases . . . . .	5
1.3 Ikke-funktionelle krav . . . . .	12
1.3.1 FURPS+ . . . . .	12
<b>Kapitel 2 Systemarkitektur</b>	<b>15</b>
2.1 Hardware . . . . .	15
2.1.1 BDD . . . . .	16
2.1.2 IBD . . . . .	17
2.2 Software . . . . .	18
2.2.1 Domænenemodel . . . . .	18
2.2.2 Klassediagram . . . . .	19
2.2.3 Sekvensdiagram . . . . .	22
<b>Kapitel 3 Design</b>	<b>27</b>
3.1 Designproces (GUI) . . . . .	27
3.2 Hardware . . . . .	28
3.3 Software . . . . .	30
<b>Kapitel 4 Produktet</b>	<b>31</b>
<b>Kapitel 5 Acceptest</b>	<b>33</b>
5.1 Funktionelle krav . . . . .	34
5.1.1 Acceptest af use cases . . . . .	34
5.2 Ikke-funktionelle krav . . . . .	44
<b>Bilag</b>	<b>47</b>
Bilag 1: Samarbejdsaftale . . . . .	48
Bilag 2: Tidsplan . . . . .	50
Bilag 3: Hardware versioner . . . . .	52
Bilag 4: Software versioner . . . . .	53
Bilag 5: Logbog . . . . .	54
Bilag 6: Mødereferater . . . . .	55
Bilag 7: Datasheet NI-6009 DAQ . . . . .	56





# Kravspekifikation

# 1

Version	Dato	Ansvarlig	Beskrivelse
0.1	9/9-15	Alle	Oprettelse af dokument
1.0	21/9-15	LB, JL, HR, RR, SV	Tilføjelse af use case " <i>Log ind</i> ", samt smårettelser efter møde med vejleder
1.1	23/9-15	Alle	Rettelser af " <i>Log ind</i> " use case, samt rettelser af andet i KS
2.0	28/9-15	Alle	Tilføjer ny use case, " <i>Kalibrer systemet</i> ", og tilretter " <i>Log ind</i> " use case
2.1	29/9-15	Alle	
3.0	7/10-15	Alle	Tilrettelser efter review med gr. 4
4.0	4/11-15	Alle	Gennemgang med ændringer efter rettelser fra vejledere

## Formål

Formålet med en kravspecifikation er, at beskrive systemets funktionelle og ikke-funktionelle krav til kunden. Kravspecifikationen er kontrakten mellem virksomhed og kunde.

## 1.1 Systembeskrivelse

Dette program skal opfylde de obligatoriske krav, opstillet af IHA:

- Programmet skal programmeres i C#
- Programmet skal kunne kalibrere blodtrykssignalet og foretage en nulpunktsjustering
- Blodtrykket skal vises kontinuert på en graf i GUI, hvor der ses systolisk og diastolisk tryk
- Målingerne skal kunne gemmes som tekstfil eller i database

- Systemet skal kunne filtrere blodtrykket i selve programmet via et digitalt filter, dette skal kunne slås til og fra.

Ud fra projektets vision, beskrevet i projektformuleringen, skal der udvikles et system til måling af blodtryk. Systemet skal kunne bruges på computere, der forudsættes at have adgang til måleudstyret, og samtidig overholder de opstillede krav.

Systemet skal kunne tilsluttes et væskefyldt kateter og vise en blodtrykskurve på en computerskærm.

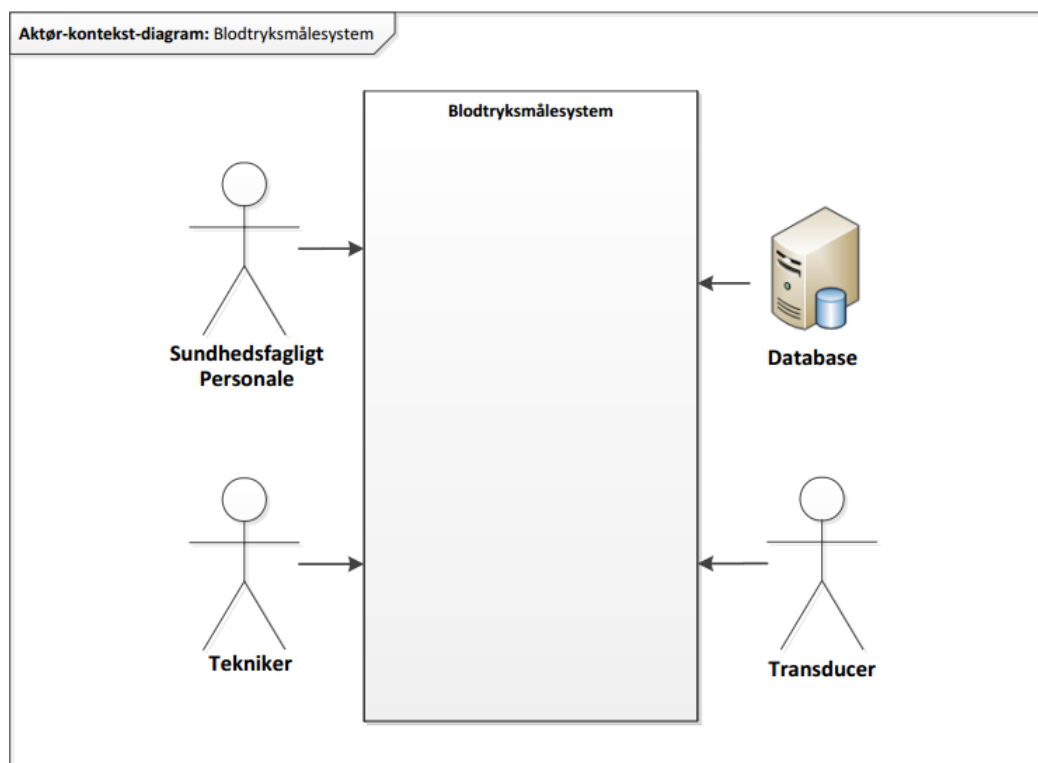
Systemet skal indeholde et elektronisk kredsløb, som forstærker signalet fra trykstransduceren og filtrerer det med et indbygget analogt filter.

Systemet skal indeholde et program, som kan vise blodtrykket som funktion af tiden. Dette foregår ved, at målingerne indlæses fra blodtryksmåleren, omdannes til et digitalt signal vha. DAQ, indlæses i et C#-program og vises grafisk.

## 1.2 Funktionelle krav

### 1.2.1 Aktør-kontekstdiagram

Der er udarbejdet et aktør-kontekst diagram med tilhørende aktørbeskrivelser, hvor de forskellige aktører i systemet er angivet og beskrevet.



Figur 1.1: Aktør-kontekstdiagram

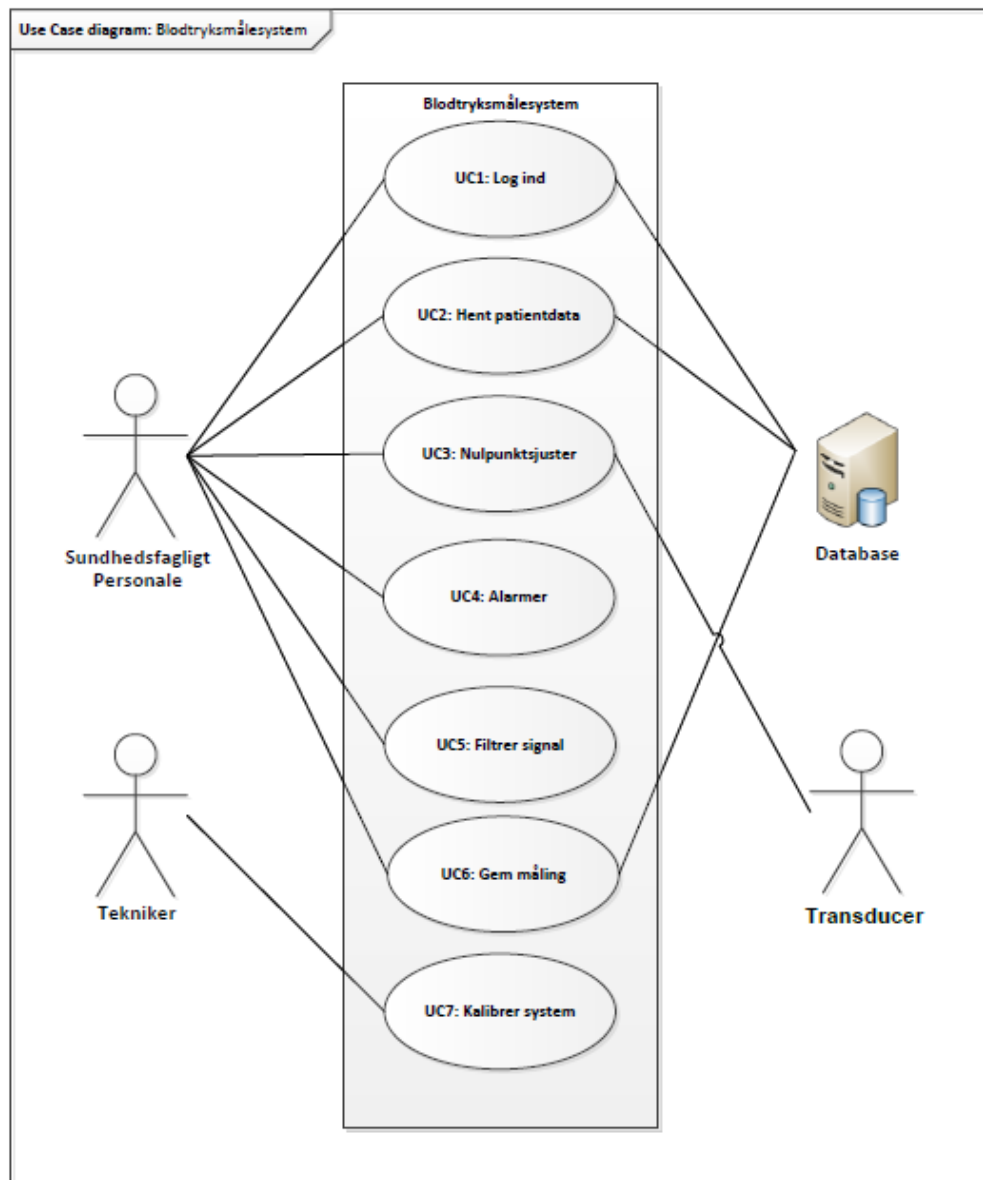
### 1.2.2 Aktørbeskrivelse

Aktørnavn	Type	Beskrivelse
Sundhedsfagligt personale	Primær	Aktøren starter, foretager og afslutter målingen. Aktøren skal have relevans i henhold til en operationsstue samt have kendskab til procedurerne herved
Tekniker	Primær	Kalibrerer systemet
Transducer	Sekundær	Transduceren omsætter tryk til et analogt elektrisk signal
Database	Sekundær	Måledataene gemmes i databasen.

*Tabel 1.2: Aktørbeskrivelse*

### 1.2.3 Use case-diagram

Der er ud fra de overordnede, definerede krav til projektet, udviklet et use case-diagram. Diagrammet viser aktørerne i systemet, samt de fire scenarier der er valgt at fokusere på i dette system.



Figur 1.2: Use case diagram

### 1.2.4 Use cases

Ud fra use case-diagrammet, er der udarbejdet en fully-dressed use case til hvert scenarie. Disse indgår herunder.

Use case 1 - Log ind	
Navn	Log ind
Scenarie	Hovedscenarie
Use case ID	UC1
Primær aktør	Sundhedsfagligt personale
Sekundær aktør(er)	Database
Initialisere	Sundhedsfagligt personale trykker på "Log ind"-knap
Mål	Sundhedsfagligt personale er logget ind og klar til at foretage måling
Forudsætninger	Systemet er operationelt
Resultat	Sundhedsfagligt personale er succesfuldt logget ind i systemet
Hovedforløb	<ol style="list-style-type: none"> <li>1. Sundhedsfagligt personale indtaster ID</li> <li>2. Sundhedsfagligt personale indtaster tilhørende password</li> <li>3. Sundhedsfagligt personale trykker på "Log ind"-knappen [3a. <i>Fejl i indtastede ID eller password</i>]</li> </ol>
Undtagelser	<ol style="list-style-type: none"> <li>3a. [<i>Fejl i indtastede ID eller password</i>]</li> <li>1. Systemet gør opmærksom på fejl, og lader bruger indtaste password og ID igen</li> <li>2. Sundhedsfagligt personale trykker på "OK" -knappen</li> </ol>

Tabel 1.3: Fully dressed Use case 1

---

**Use case 2 - Hent patientdata**


---

Navn	Hent patientdata
Scenarie	Hovedscenarie
Use case ID	UC2
Primær aktør	Sundhedsfagligt personale
Sekundær aktør(er)	Database
Initialisere	Efter UC1 er kørt succesfuldt
Mål	At indhente patientens personlige oplysninger og vise patientens CPR-nummer på GUI'en
Forudsætninger	UC1 er kørt succesfuldt
Resultat	Patientens personlige oplysninger er hentet og CPR-nummeret vises i GUI'en
<hr/>	
Hovedforløb	<ol style="list-style-type: none"> <li>1. Sundhedsfagligt personale indtaster patientens CPR-nummer</li> <li>2. Sundhedsfagligt personale trykker på knappen "<i>Hent patientoplysninger</i>" [2a. <i>Det indtastede CPR-nummer er ikke gyldigt</i>]</li> <li>3. Patientens CPR-nummer vises i GUI'en</li> </ol>
<hr/>	
Undtagelser	<ol style="list-style-type: none"> <li>2a. [<i>Det indtastede CPR nummer er ikke gyldigt</i>]</li> <li>1. Systemet gør bruger opmærksom på fejl, og beder om ny indtastning af CPR nummer</li> </ol>

---

Tabel 1.4: Fully dressed Use case 2

Use Case 3 - Nulpunktsjuster	
Navn	Nulpunktsjuster
Scenarie	Hovedscenarie
Use case ID	UC3
Primær aktør	Tekniker
Sekundær aktør(er)	Transducer
Initialisere	Systemet
Mål	At nulpunktsjustere systemet i forhold til det aktuelle tryk samt at blodtryksmålingen starter og vises grafisk
Forudsætninger	UC1 og UC2 er kørt succesfuldt
Resultat	Systemet er nulpunktsjusteret i forhold til det aktuelle tryk og blodtryksmålingen starter og vises grafisk
Hovedforløb	<ol style="list-style-type: none"> <li>1. Sundhedsfagligt personale trykker på "Nulpunktsjustering" -knappen</li> <li>2. Nulpunktsjustering foretages</li> <li>3. Blodtrykket vises i GUI'en [3a. <i>Blodtryk for højt eller lavt</i>]</li> </ol>
Undtagelser	<ol style="list-style-type: none"> <li>3a. [<i>Blodtryk for højt eller lavt</i>]</li> <li>1. UC4 startes</li> </ol>

Tabel 1.5: Fully dressed Use case 3

---

**Use Case 4 - Alarmer**


---

Navn	Alarmer
Scenarie	Hovedscenarie
Use case ID	UC4
Primær aktør	Sundhedsfagligt personale
Sekundær aktør(er)	
Initialisere	UC3 undtagelse 3a
Mål	At alarmere sundhedsfagligt personale om for højt eller lavt blodtryk
Forudsætninger	UC1, UC2 og UC3 er kørt succesfuldt
Resultat	Sundhedsfagligt personale er alarmeret om for højt eller lavt blodtryk
Hovedforløb	<ol style="list-style-type: none"> <li>1. Systemet alarmerer sundhedsfagligt personale med lyd</li> <li>2. Sundhedsfagligt personale har nu mulighed for at slå systemets alarm på "<i>Lydløs</i>" -tilstand i en periode på tre minutter</li> <li>3. Alarmen stopper ved normalisering af blodtrykket</li> </ol>

---

*Tabel 1.6: Fully dressed Use case 3*



---

**Use Case 5 - Filtrer signal**


---

Navn	Filtrer signal
Scenarie	Hovedscenarie
Use case ID	UC5
Primær aktør	Sundhedsfagligt personale
Sekundær aktør(er)	
Initialisere	Sundhedsfagligt personale
Mål	At filtrering af signalet er slået til/fra
Forudsætning	UC2 er kørt succesfuldt
Resultat	Filtreringen af signalet er slået til/fra
Hovedforløb	<ol style="list-style-type: none"> <li>1. Sundhedsfagligt personale trykker på "Fra" - knappen</li> <li>2. Sundhedsfagligt personale trykker på "Til" - knappen</li> </ol>

---

*Tabel 1.7: Fully dressed Use case 5*

---

**Use Case 6 - Gem data**


---

Navn	Gem data
Scenarie	Hovedscenarie
Use case ID	UC6
Primær aktør	Sundhedsfagligt personale
Sekundær aktør(er)	Database, tekniker
Initialisere	Sundhedsfagligt personale
Mål	At gemme måledataene i en database
Forudsætninger	UC2 og UC3 er gennemført
Resultat	Måledata er gemt korrekt i databasen
Hovedforløb	<ol style="list-style-type: none"> <li>1. Sundhedsfagligt personale trykker på "<i>Gem data</i>" - knappen</li> <li>2. Måledata gemmes i databasen [2a. <i>Måledata kan ikke gemmes</i>]</li> <li>3. Systemet giver beskeden: "<i>Data gemt</i>"</li> </ol>
Undtagelser	<ol style="list-style-type: none"> <li>2a. [<i>Måledata kan ikke gemmes</i>]</li> <li>1. Der kommer en pop-up meddelelse "<i>Data er ikke gemt - tekniker er tilkaldt</i>"</li> <li>2. Sundhedsfagligt personale trykker "<i>OK</i>"</li> </ol>

---

*Tabel 1.8: Fully dressed Use case 6*

---

**Use Case 7 - Kalibrer system**


---

Navn	Kalibrer system
Scenarie	Hovedscenarie
Use case ID	UC7
Primær aktør	Tekniker
Sekundær aktør(er)	
Initialisere	Systemet
Mål	At justere systemet i forhold til kalibrering
Forudsætninger	Tekniker er tilkaldt
Resultat	Systemet er justeret
Hovedforløb	<ol style="list-style-type: none"> <li>1. Tekniker påtrykker systemet tre kendte tryk</li> <li>2. Tekniker aflæser responserne på GUI</li> <li>3. Tekniker noterer afvigelserne fra de kendte tryk [3a. <i>Der er ingen afvigelser</i>]</li> <li>4. Tekniker justerer afvigelsen i systemets software</li> <li>5. UC5 startes forfra</li> </ol>
Undtagelser	<ol style="list-style-type: none"> <li>3a. [<i>Der er ingen afvigelse</i>]</li> <li>1. UC5 afsluttes</li> </ol>

---

*Tabel 1.9: Fully dressed Use case 7*

## 1.3 Ikke-funktionelle krav

Ikke-funktionelle krav beskrevet ved FURPS+ med MoSCoW.

### 1.3.1 FURPS+

MoSCoW er angivet i en parentes med enten M, S, C eller W.

#### Functionality

1. (M) Programmet skal programmeres i C#, Visual Studio
2. (S) Systemet bør kunne angive pulsen via en lyd ved hvert hjerteslag ved .... Hz
3. (M) Blodtrykket skal kunne gemmes i en database og skal indeholde
  - a) Patient-CPR, ansvarligt sundhedspersonale, ansvarlig organisation, dato
  - b) Rådata, samplerate (Hz), interval (s), data format, måleformat, starttid, antal målinger
4. (M) Blodtrykket skal måles indenfor 10 mmHg præcision
5. (M) Forstærkeren skal kunne forstærke det elektriske signalet
6. (M) Lavpasfilteret skal kunne filtrere støj fra det elektriske signal

#### Usability

7. (M) Programmet skal indeholde en "*Log ind*" -knap
8. (M) Programmet skal indeholde en "*Hent patientoplysninger*" -knap
9. (M) Programmet skal indeholde en "*Nulpunktsjustering*" -knap
10. (M) Programmet skal indeholde en "*Lydløs*" -knap
11. (M) Programmet skal indeholde en "*Gem data*" -knap
12. (M) Programmet skal indeholde en "*Til*" -radiobutton
13. (M) Programmet skal indeholde en "*Fra*" -radiobutton
14. (M) Programmet skal indeholde en "*OK*" -knap i pop-up vinduet

#### Reliability

15. (S) Systemet bør kunne køre fejlfrit i et år
16. (S) Systemet bør have en "mean time to restore" på højst 24 timer  
Systemet får herved en tilgængelighed beregnet ved

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

MTBF = "mean time between failure"

MTTR = "mean time to restore"

**Performance**

17. (M) Systemet skal kontinuert vise en grafisk afbildning af blodtrykket, hvor tryk er op af y-aksen og tiden er på x-aksen i intervallet af 6 sekunder

**Supportability**

18. (S) Softwaren bør være opbygget af trelagsmodellen  
+ **Test conditions**



# Systemarkitektur 2

Version	Dato	Ansvarlig	Beskrivelse
0.1	4/11-15	Alle	Tilføjelse af arkitektur
Tekst	Tekst	Tekst	Tekst
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.

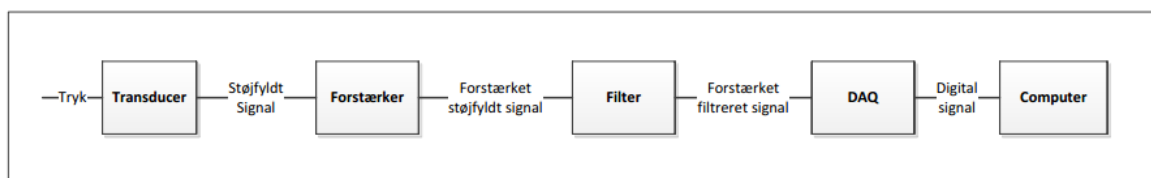
## Formål

Til beskrivelse af systemarkitekturen og det detaljerede design for produktet, er der benyttet SysML. SysML anvendes her, da blodtryksmålesystemet både indeholder software og hardware. Et af de vigtigste argumenter for brug af SysML er, at de fastlagte standarder i sproget medfører en bedre formidling af systemet, hvilket giver et større overblik.

## 2.1 Hardware

Hardware-delen består af et elektronisk kredsløb, som forstærker signalet fra tryktransduceren og filtrerer det med et indbygget analogt filter.

Til at skabe overblik over blodtryksmålesystemets hardware er der uarbejdet en figur, der viser hele det overordnede system.



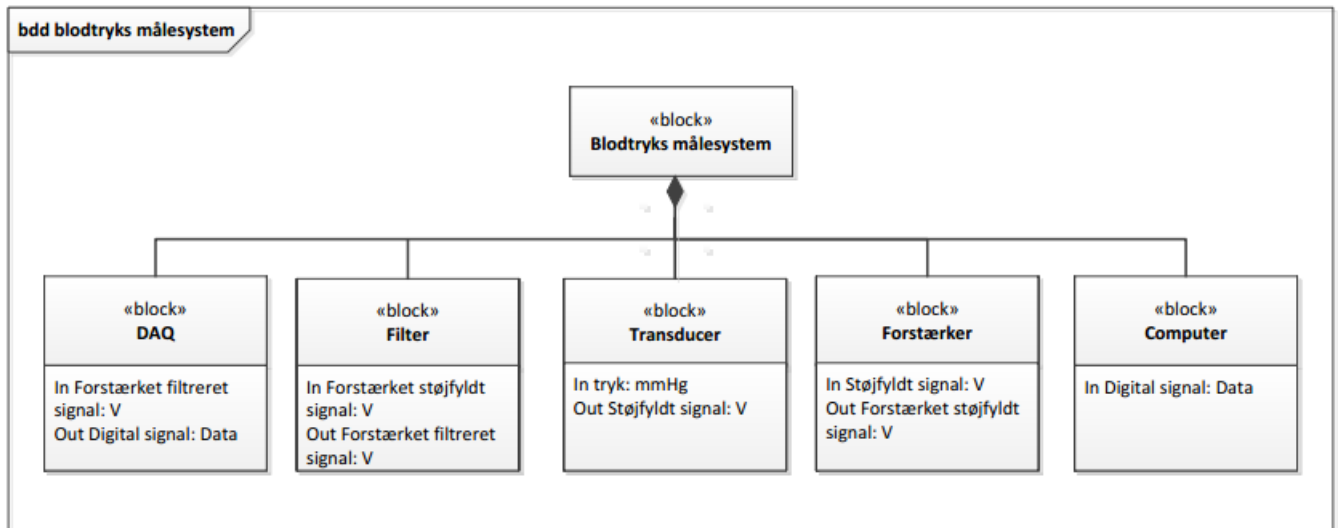
Figur 2.1: Blodtryksmålersystemet

Denne illustrerer, at der ind i transduceren kommer tryk og ud kommer et støjfyldt signal. Dette signal bliver ved forstærkeren forstærket og heraf et forstærket støjfyldt signal. Igennem filtret bliver støjen filtreret fra. Det filtrerede signal føres igennem DAQ'en, som omdanner det til et digitalt signal, som anvendes i computerens softwareprogram.

Til at præcisere komponenterne i blodtryksmålesystemets hardware, er der valgt at lave strukturdiagrammer. Her er der anvendt blokdefinitionsdiagram (BDD) og et internt blokdiagram (IBD).

### 2.1.1 BDD

BDD'et er anvendt til, at dokumentere nedbrydningen af systemet og forholdene mellem blokkene.



Figur 2.2: BDD

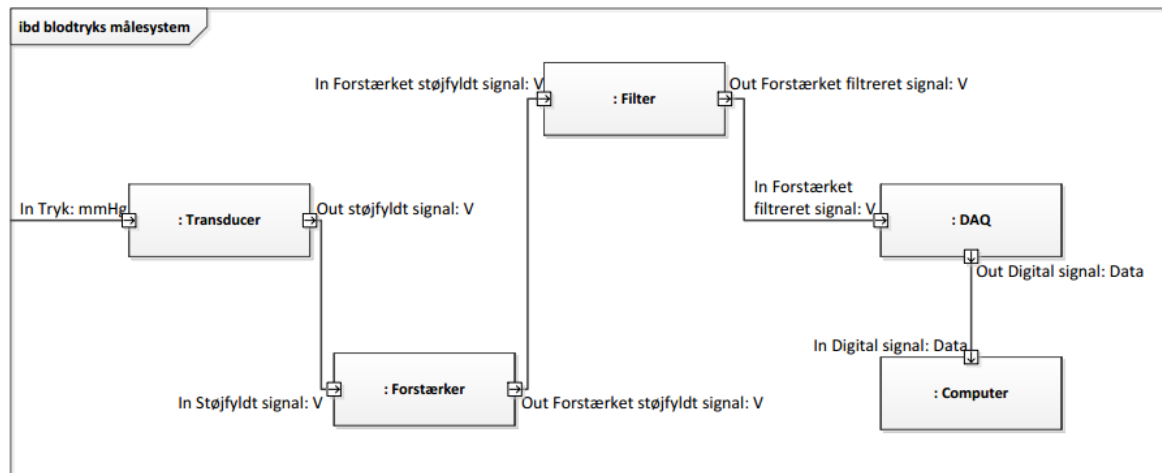
#### Blokbeskrivelser:

- Transducer: En tryktransducer, som konverterer et tryk til et analogt elektrisk signal
- Forstærker: Signalet forstærkes således at hele forsyningsspændingen udnyttes
- Filter: Et 2. ordens lavpasfilter fjerner højfrekvent støj
- DAQ: A/D konverter omsætter den analoge indgangsspænding til et digitalt signal
- Computer: Enheden som indeholder softwareprogrammet til visning af blodtryk



### 2.1.2 IBD

IBD'et er anvendt til, at dokumentere den interne struktur i blokkene.



Figur 2.3: IBD

#### Signalbeskrivelse

Forbindelse	Signaltype	Funktionalitet
Transducer - forstærker	Støjfyldt signal	Elektrisk analogt signal med støj i enheden volt
Forstærker - filter	Forstærket støjfyldt signal	Elektrisk analogt forstærket støjfyldt signal i enheden volt
Filter - DAQ	Forstærket filtreret signal	Elektrisk analogt forstærket filtreret signal i enheden volt
DAQ - computer	Digitalsignal	Elektrisk digitalt signal med data via USB
Batterier - transducer, forstærker, filter	Forsyningsspænding	Positiv og negativ 9V

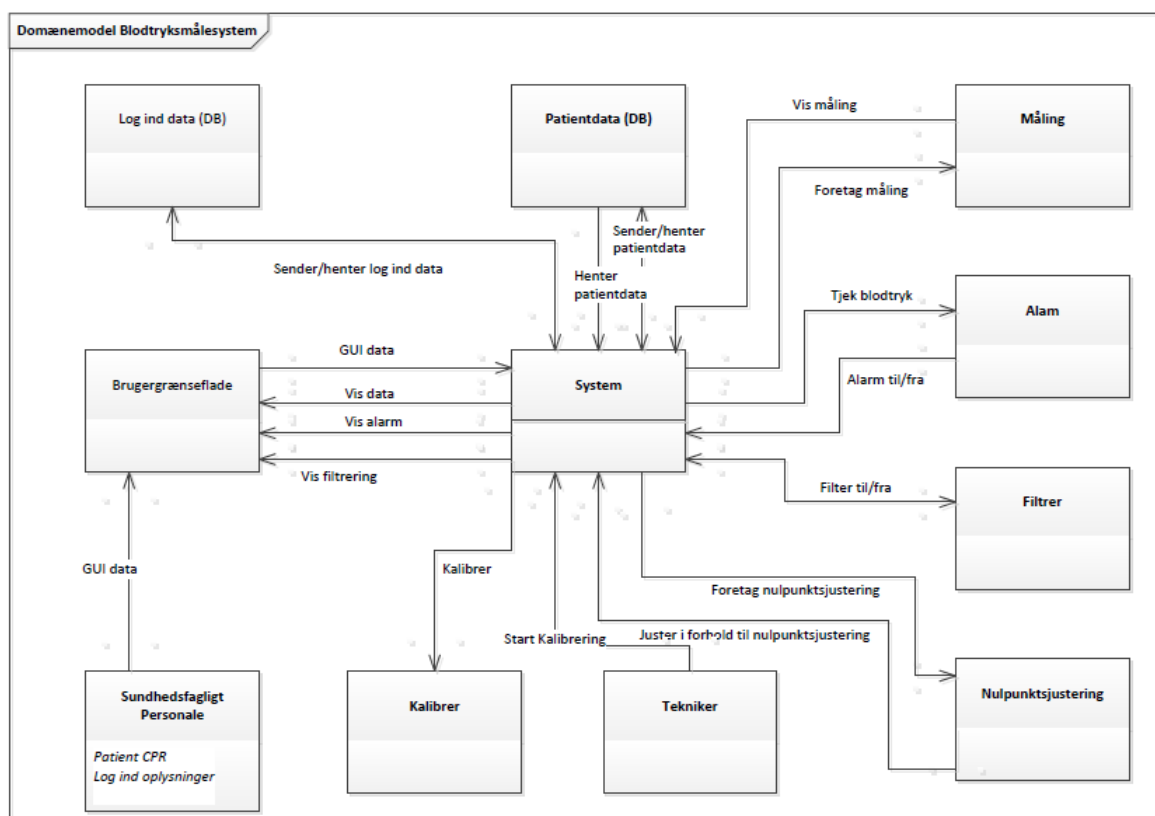
## 2.2 Software

Brugergrænsefladen i software-delen består af to forskellige GUI'er, en til at logge ind og en til diagnostik. Programmet indeholder en række klasser indeholdende funktionaliteten beskrevet i UC's samt databaser til opbevaring af data. Softwaren er opbygget af trelagsmodellen.

For at skabe et overblik over sammenhængen mellem UC's og softwaren i systemet, er der udviklet en applikationsmodel. Applikationsmodellen indeholder en domænemodel over hele systemet, et klassediagram for hver enkelt UC, et sekvensdiagram over hele systemet, et sekvensdiagram for hver UC og et opdateret klassediagram med metoder. Ved at opdele de forskellige dele i softwaren samt at oprette klasser efter den ønskede funktionalitet i UC's, opnås en sammenhænge og overskuelighed over systemet som helhed.

### 2.2.1 Domænemodel

Domænemodellen er udviklet vha. navneordsanalyse i de fem UC's. Domænemodellen giver et overblik over hvilken funktionalitet der - ud fra UC's - er relevant. Funktionaliteterne er opdelt i kasser, der senere bliver til klasser i softwaren.



Figur 2.4: Domænemodel blodtryksmålssystem

### 2.2.2 Klassediagram

Klassediagrammerne for hver enkelt UC viser sammenhængen mellem de forskellige instanser i den enkelte UC.

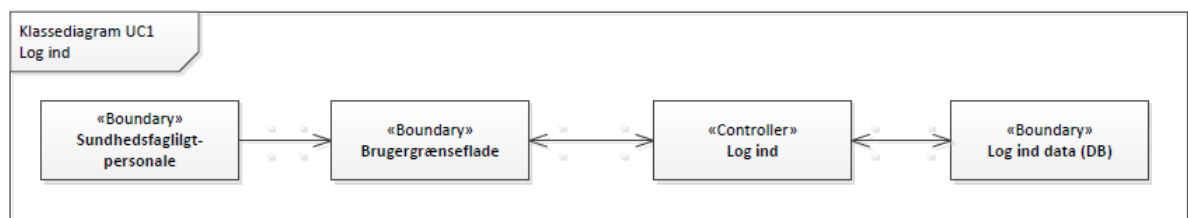
*Boundary-klasser* er den aktuelle UC's aktører.

*Controller-klassen* indeholder UC'ens funktionalitet og udfører UC'en ved at interagere med boundary-klasserne og domain-klasserne. Controller-klassen er opkaldt efter den aktuelle UC's navn.

*Domain-klassen* repræsenterer systemets domæne og hukommelse.

#### Klassediagram UC1

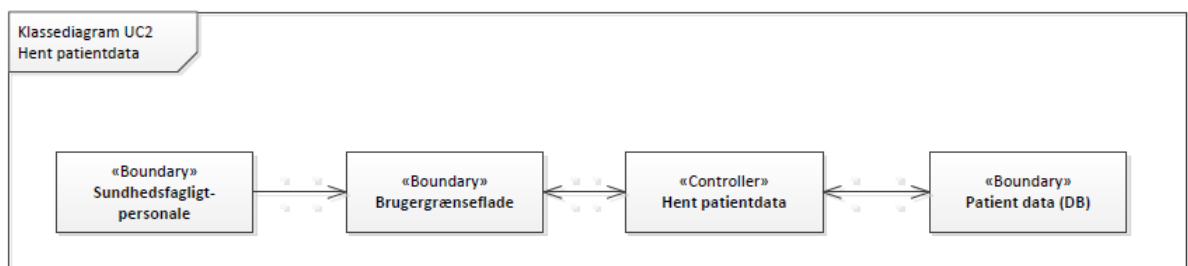
I klassediagrammet for UC1 logger sundhedsfagligt personale ind vha. brugergrænsefladen - de er begge to boundary-klasser. Brugergrænsefladen sender besked til controller-klassen "Log ind". Log ind-data hentes - via controlleren - i Log ind databasen og sendes tilbage til brugergrænsefladen via controlleren, hvorved sundhedsfagligt personale logges ind.



Figur 2.5: Applikationsmodel UC1

#### Klassediagram UC2

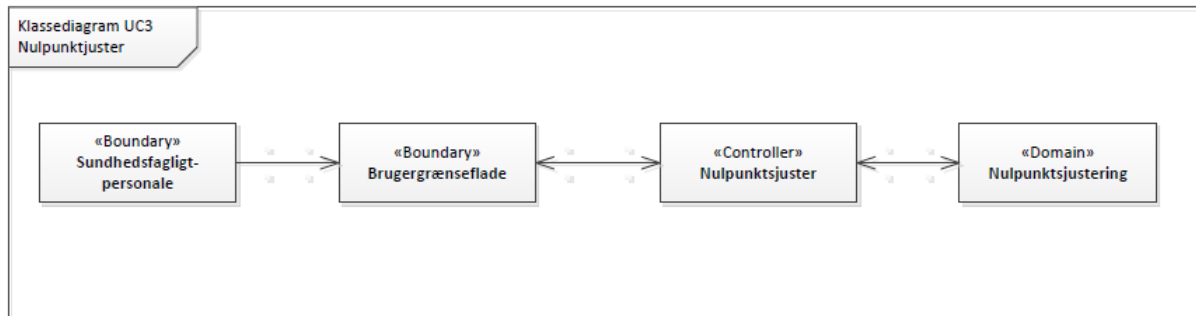
I klassediagrammet for UC2 er sundhedsfagligt personale og brugergrænsefladen boundary-klasser. "Hent patientoplysninger" er UC'ens controller-klasse. Patientoplysningerne hentes fra Patient databasen - en boundary-klasse - og sendes tilbage igennem controller-klassen til brugergrænsefladen.



Figur 2.6: Applikationsmodel UC2

### Klassediagram UC3

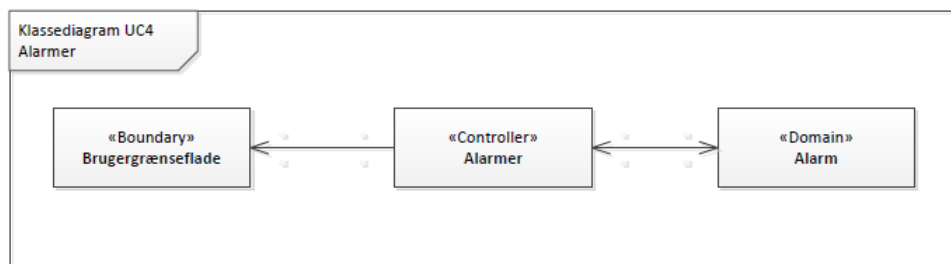
I klassediagrammet for UC3 er sundhedsfagligt personale og brugergrænsefladen boundary-klasser. "Nulpunktsjuster" er UC'ens controller-klasse, som udfører UC'en vha. domain-klassen "Nulpunktsjustering". Nulpunktsjusteringen sendes tilbage til brugergrænsefladen fra domain-klassen, igennem controller-klassen.



Figur 2.7: Applikationsmodel UC3

### Klassediagram UC4

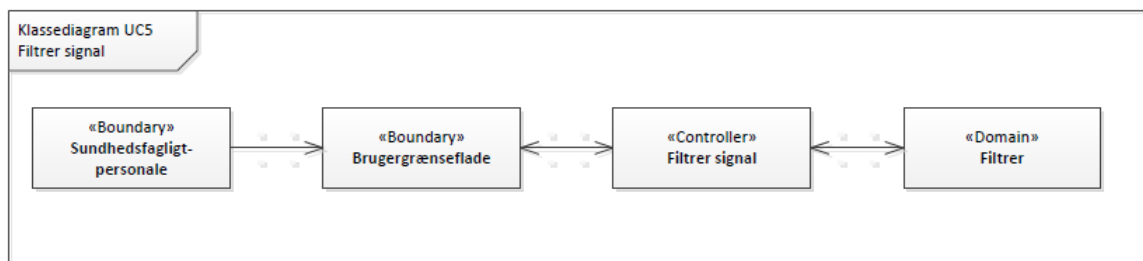
I klassediagrammet for UC4 er brugergrænsefladen boundary-klasse. "Alarmer" er UC'ens controller-klasse, som sender besked omkring alarmering til domain-klassen "Alarm". Domain-klassen alarmerer, og sender alarmen til brugergrænsefladen via controller-klassen.



Figur 2.8: Applikationsmodel UC4

### Klassediagram UC5

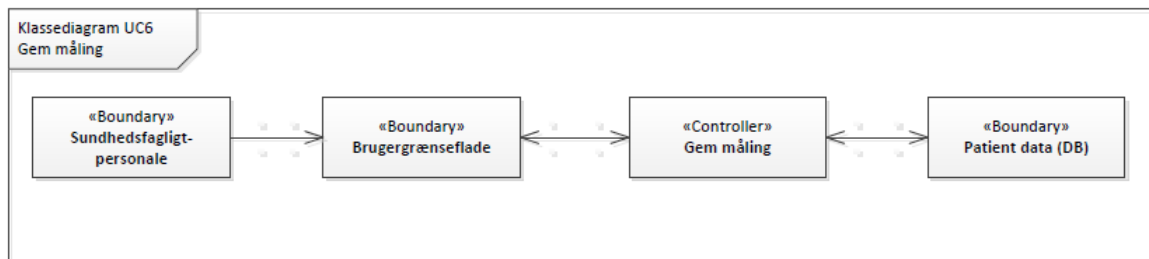
Klassediagrammet for UC5 viser, at sundhedsfagligt personale og brugergrænsefladen er boundary-klasser. "Filtrer signal" er UC'ens controller-klasse som får besked fra brugergrænsefladen om at filtrere signalet - og sender filtreringen tilbage til brugergrænsefladen.



Figur 2.9: Applikationsmodel UC5

### Klassediagram UC6

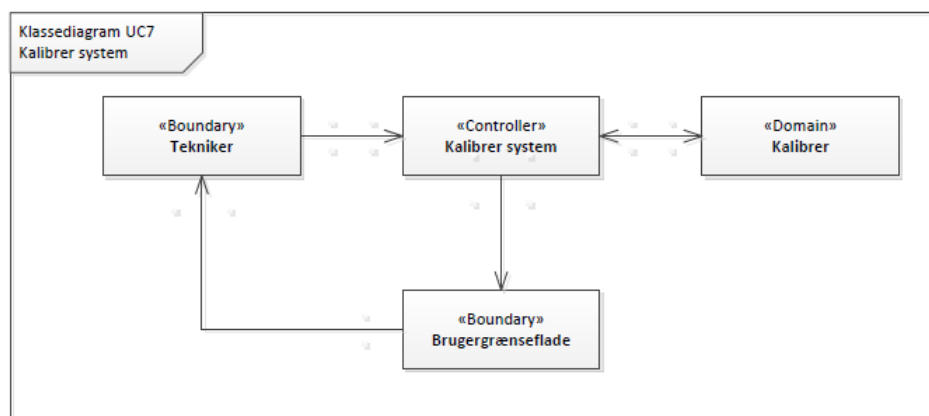
I klassediagrammet for UC5 er sundhedsfagligt personale og brugergrænsefladen boundary-klasser. "Gem måling" er UC'ens controller-klasse, som får besked fra brugergrænsefladen om at gemme den akutte måling. Denne information sendes videre til domain-klassen "Patientdata" som sender besked tilbage til brugergrænsefladen, igennem controller-klassen om, at data er gemt.



Figur 2.10: Applikationsmodel UC6

### Klassediagram UC7

I klassediagrammet for UC6 ses det, at tekniker og brugergrænsefladen er boundary-klasser. Teknikeren kalibrerer systemet i controller-klassen, som justerer brugergrænsefladen i forhold til kalibreringen.



Figur 2.11: Applikationsmodel UC7

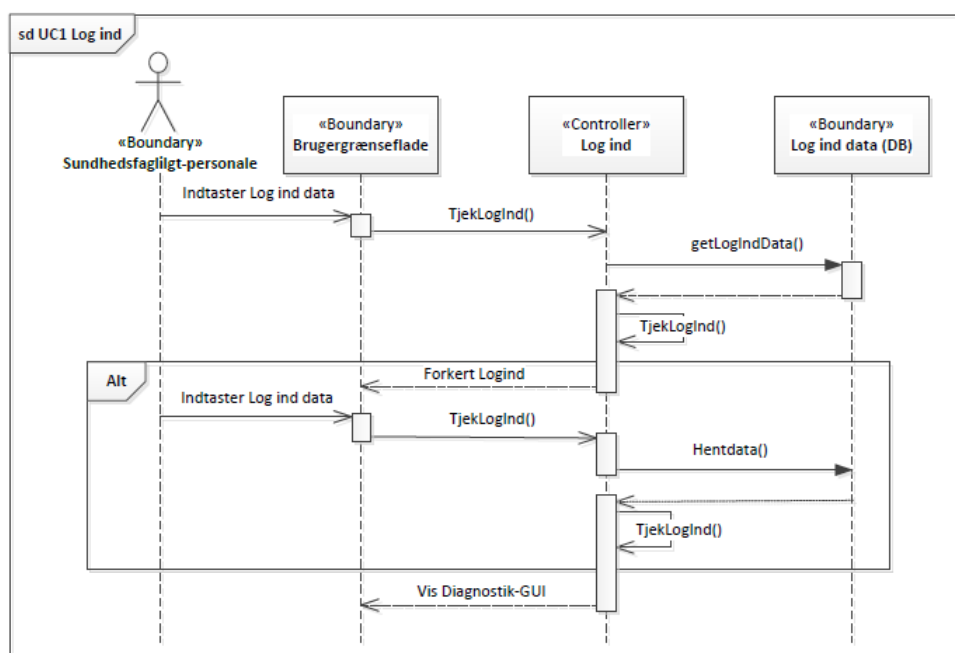
### 2.2.3 Sekvensdiagram

Controlleren i sekvensdiagrammerne er logiklaget, hvor domainklasserne er klasser tilkoblet logiklaget.

#### Sekvensdiagram UC1

Sekvensdiagrammet for UC1 viser, at sundhedsfagligt personale indtaster Log ind data i brugergrænsefladen. Disse behandles - vha. metoden TjekLogInd() - i kontrolleren Log Ind. Dataen hentes i databasen - vha. getLogIndData() - Log Ind data og sendes tilbage til kontrolleren og tjekkes heri. Er oplysningerne korrekte, vises Diagnostik-GUI'en.

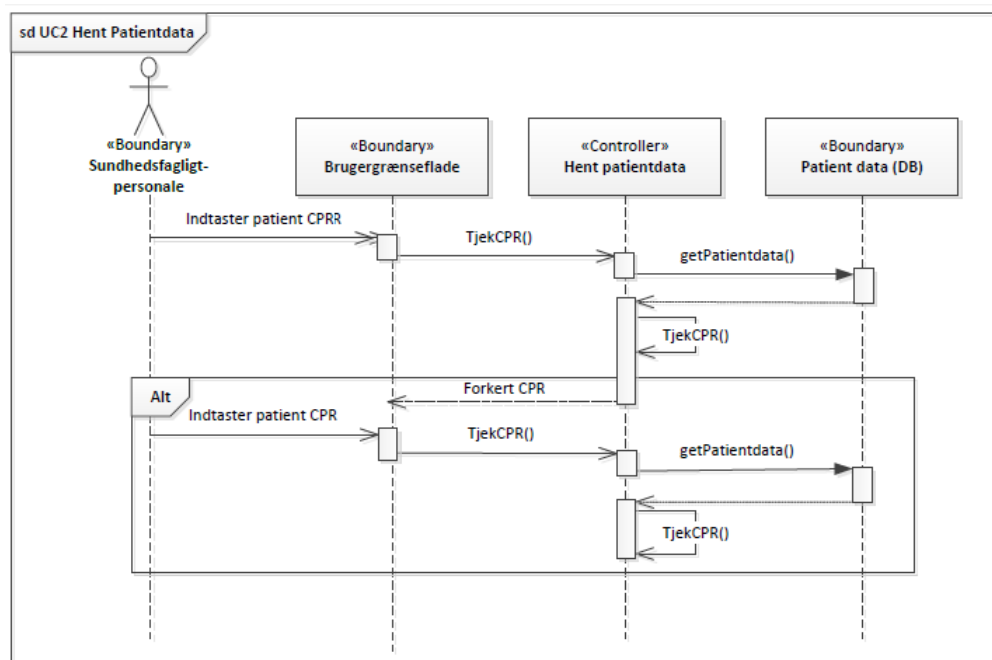
Der er en alternativ rute, hvor de indtastede oplysninger er forkerte. Dette vises i LogInd-GUI'en, og bruger indtaster oplysningerne forfra.



Figur 2.12: Sekvensdiagram UC1

#### Sekvensdiagram UC2

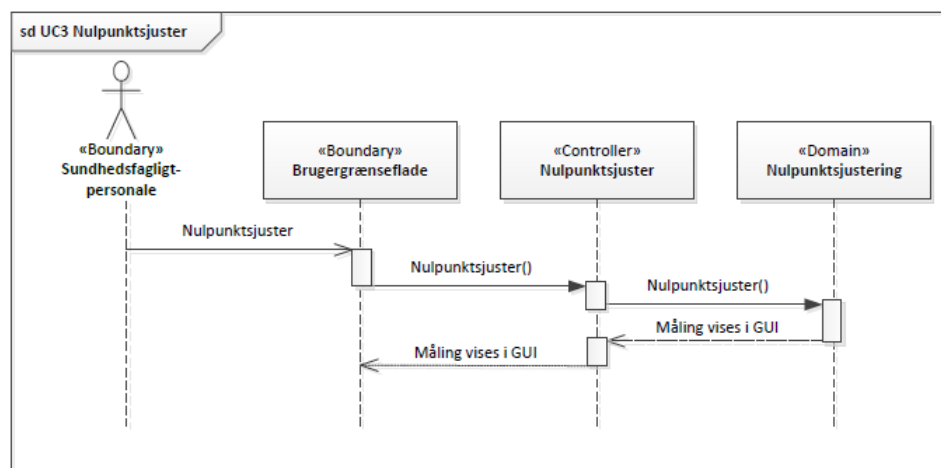
I diagrammet for UC2 ses det, at sundhedsfagligt personale indtaster patientens CPR i brugergrænsefladen. Patienten tjekkes i kontrolleren vha. TjekCPR(). Data hentes fra Patient data - vha. getPatientdata() - og sendes tilbage til kontrolleren, som tjekker disse. Den alternative rute består i, at det indtastede CPR er forkert. I det tilfælde, sendes en besked til brugeren herom, og der kan ske en ny indtastning.



Figur 2.13: Sekvensdiagram UC2

### Sekvensdiagram UC3

Diagrammet for UC3 viser at sundhedsfagligt personale, igennem brugergrænsefladen, beder om at få systemet nulpunktsjusteret. Nulpunktsjusteringen sendes som besked fra brugergrænsefladen til kontrolleren via metoden NulpunktsJuster(). Denne metode sendes videre til Nulpunktsjustering-klassen, som sender en besked, igennem kontrolleren, til brugergrænsefladen om, at blodtryksmålingen skal vises i brugergrænsefladen.



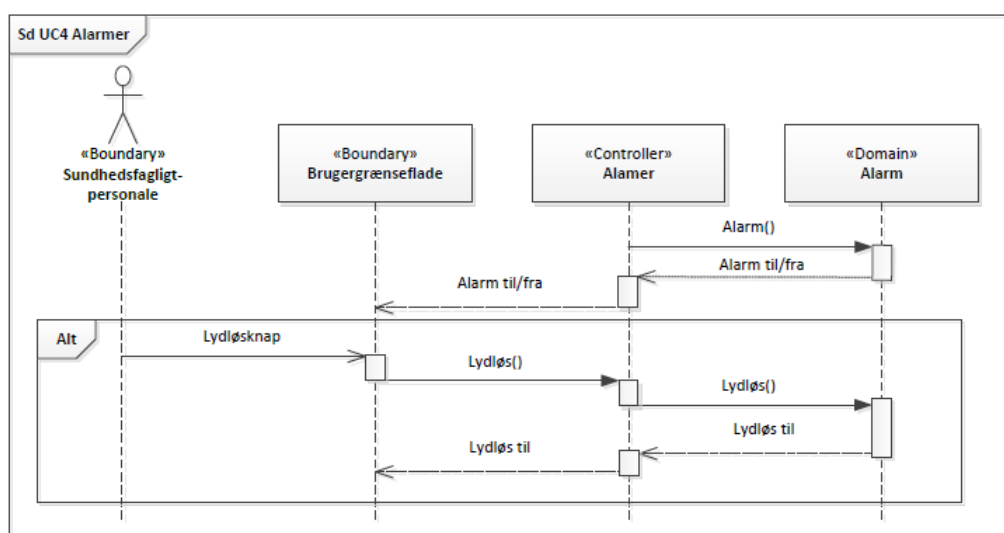
Figur 2.14: Sekvensdiagram UC3

### Sekvensdiagram UC4

I diagrammet beder kontrolleren ved hjælp af metoden Alarm() om at Alarm-klassen skal tjekke op på om alarmfunktionen skal slås til eller fra. Efter metoden Alarm() er kørt gives der besked tilbage til kontrolleren og videre til brugergrænseflade, hvor man ville kunne se

om amarmen er slået til eller fra. Efterfølgende kan der forekomme en alternativ funktion, hvor brugeren ved hjælp af lydløsknappen på brugergrænsefladen kan slå lydløs funktionen til. Når brugeren har trykket på knappen, starter controlleren metoden `Lydløs()`, som sender videre besked om at blive kørt i Alarm-klassen. Der sendes en returbesked tilbage til gennem controlleren og videre til brugergrænsefladen.

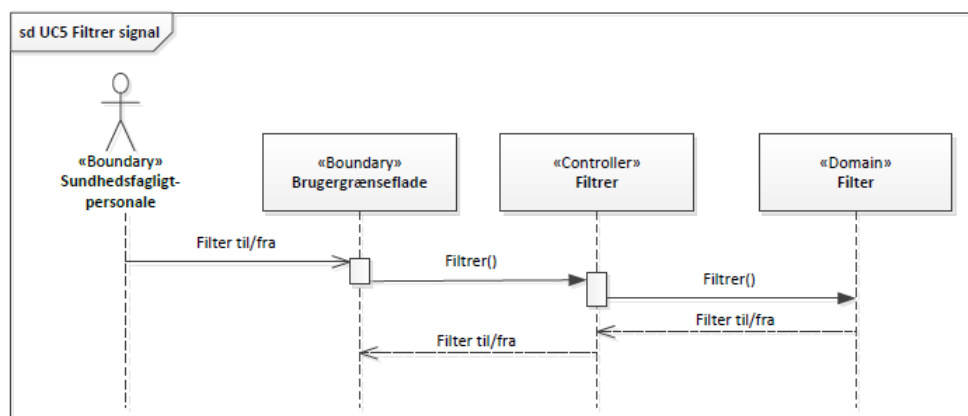
I sekvensdiagrammet over UC4 ses det, at det er controlleren der sætter alarmen igang så snart blodtryksværdierne er udenfor de angivne grænseværdier.



Figur 2.15: Sekvensdiagram UC4

## Sekvensdiagram UC5

Sundhedsfagligt personale trykker på til/fra-knappen i brugergrænsefladen. Metoden `Filtrer()` i controllerklassen giver besked om at `Filtrer()` skal køres i Domænklassen "Filter". Der gives efterfølgende besked til controllerklassen og videre til brugergrænsefladen, hvor der kan ses ved hjælp af en radiobutton om filteret er slået til eller fra.

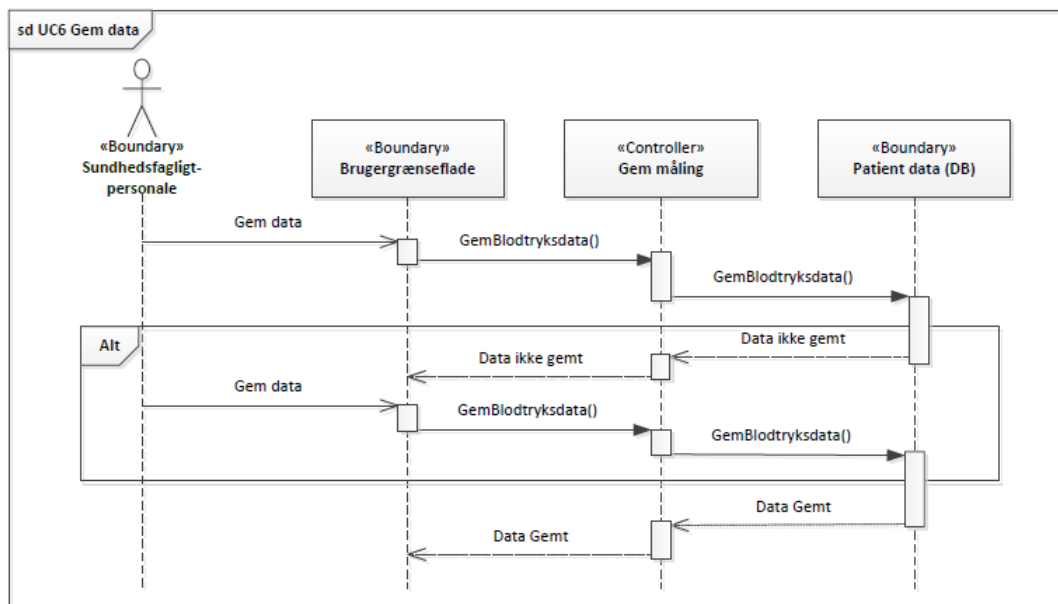


Figur 2.16: Sekvensdiagram UC5



### Sekvensdiagram UC6

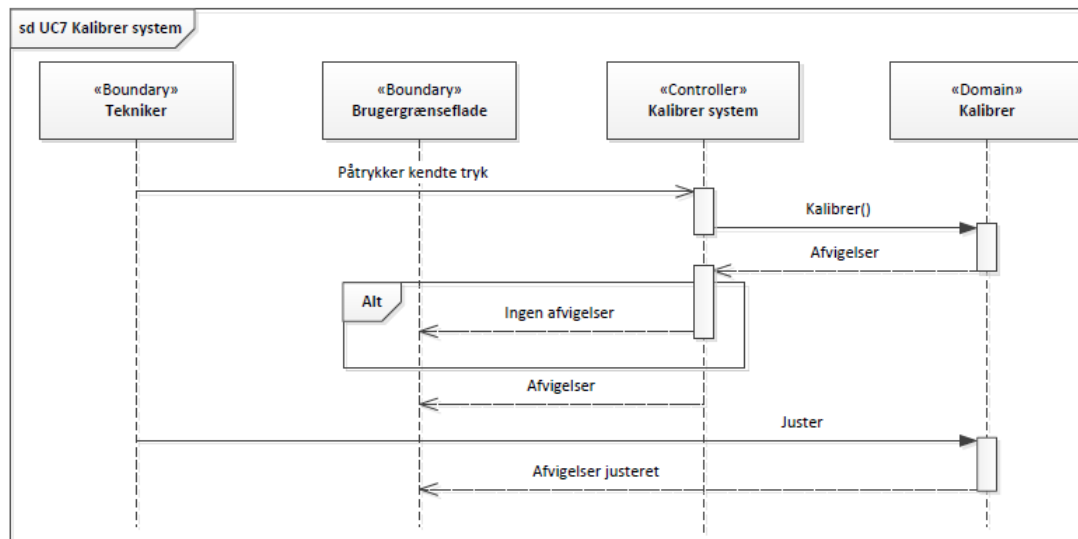
Sundhedsfagligt personale trykker på gem knap i brugergrænsefladen. Dette aktiverer metoden `GemBlodtryksdata()` i controlleren, som giver videre besked til at køre den i boundaryklassen "Patient Data (DB)". Hvis der sker en fejl under gemmeprocessen og målingen dermed ikke bliver gemt, vil et alternativt handlingsforløb indtræde. "Patient data (DB)" vil i dette tilfælde give en returbesked til controlleren og videre til brugergrænsefladen, som meddeler til det sundhedsfaglige personale at målingen ikke er gemt. Herefter vil det sundhedsfaglige personale kunne prøve igen. Er målingen gemt, vil der blive givet besked til controlleren fra "Patient data (DB)", som videre vil give besked til brugergrænsefladen.



Figur 2.17: Sekvensdiagram UC6

### Sekvensdiagram UC7

I diagrammet ses det at teknikeren starter med at påtrykke et kendt tryk i controlleren. Dette vil starte metoden `Kalibrer()` i Domainklassen "Kalibrer". "Kalibrer" giver besked om afvigelser til controlleren. I tilfælde at der ikke er nogen afvigelser gives der besked om dette og teknikeren fortager ikke nogen ændringer. Ved afvigelser justere teknikeren det ved hjælp af "Kalibrer"-klassen og der gives direkte besked tilbage til brugergrænsefladen om ændringerne.

*Figur 2.18: Sekvensdiagram UC7*

Version	Dato	Ansvarlig	Beskrivelse
0.1	6/11-15	SV	Tilføjelse af arkitektur
0.2	11/11-15	SV	Tilføjelse af design
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.

**Formål** I dette kapitel er design af SW og HW illustreret med figurer og forklaret.

## 3.1 Designproces (GUI)

I udvikling af projektets brugergrænseflader er der brugt generelle principper om gode brugergrænseflader, MMI. Det er specielt prioriteret at brugeren skal være i kontrol, og at brugerens sprog skal være det gennemgående, da sproget gerne skal ligge til brugerens logik og ikke udviklerens. Brugergrænsefladen er udviklet med sigende knapper, radiobuttons, labels og tekstbokse.

Efter et besøg på dagkirurgisk afsnit på Aarhus Universitets Hospital, Skejby, er brugergrænsefladen forsøgt udviklet således at den passer til brug på en operationsstue. Ideelt set skulle der være to forskellige skærme, og dermed to forskellige brugergrænseflader - én til indtastning af brugeroplysninger og patientdata, og én til diastolisk,- systolisk- og pulsmåling. Brugergrænsefladen med patientoplysninger er til anæstesisygeplejerske/læge hvor brugergrænsefladen indeholdende målinger er til kirurgen. Kirurgens brugergrænseflade skal være simpel og indeholde så få oplysninger som muligt. Denne skal også have en alarmeringsfunktion, som kan sættes på lydløs i eksempelvis tre minutter. Anæstesisygeplejerskens brugergrænseflade skal indeholde mange (nødvendige) informationer om patienten, og skal derfor opbygges så dette er overskueligt.

Ud fra den erfarede viden på Dagkirurgisk Afsnit, er det besluttet at sammensmelte de to omtalte skærme, men dog udvikle to forskellige brugergrænseflader. Den første brugergrænseflade indeholder bruger log ind, og en fejlmelding hvis de indtastede data er forkerte. Den anden brugergrænseflade indeholder patientdata samt målingen med de nødvendige informationer og funktioner. For at brugergrænsefladerne skal kunne udgøres for de to skærme på operationstuen, er disse gjort overskuelige og selvsigende.

*Feedback to user*

Feedback til brugeren skal gives for, at bruger kan se om en evnetuel kommando er forstået. Hvis ikke der er noget feedback efter en given kommando, ved brugeren ikke, om kommandoen er forstået eller accepteret - og tror dermed, at der er fejl i systemet. En sådan feedback skal have kort reponstid og/eller et *arbejder*-symbol som viser at systemet bearbejder kommandoen.

Feedback til brugeren er implementeret således, at der eksempelvis kommer en pop-up meddelelse ved forkert log ind. Når bruger logger ind, skifter brugergrænsefladen hurtigt, så brugeren ved, at handlingen er accepteret. ....

*Never interrupt the user*

Brugeren skal aldrig forstyrres unødvendigt. Eksempelvis pop-up vinduer, som bruger ikke selv har bedt om, er forstyrrende. De forstyrrende elementer flytter brugers fokus, sætter bruger ud af kontrol og bryder brugers koncentration. Dog kan advarsler være vigtige, og dem af helt vigtig karakter, kan være nødvendig som pop-up. Advarsler af mindre vigtig karakter, kan opstå som pop-up ikoner i et hjørne - uden brug af lyde, pop-up eller andet - så bruger kan reagere når hen ikke længere er fuldt optaget.

HVORDAN ER DET IMPLEMENTERET?

*The user should be in control*

Brugeren skal lave kommandoerne og systemer skal adlyde disse, og ikke omvendt. Dette kan eksempelvis være en dialogbox på computer med knapperne "OK" til at acceptere, "Cancel" hvis bruger fortryder og "Help" hvis bruger har brug for hjælp.

HVORDAN ER DET IMPLEMENTERET?

*Speak the users language*

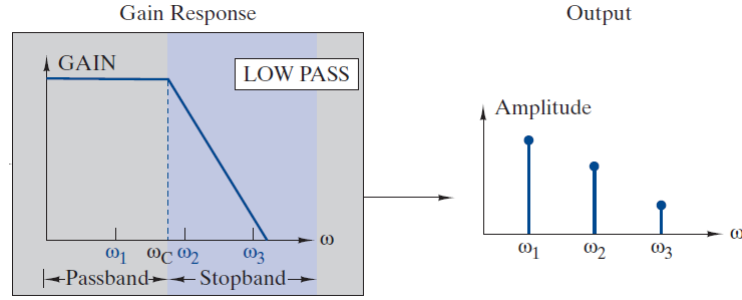
*Design should reflect the user's logic, not the constructor's logic*

*The design of a button should reflect its importance*

## 3.2 Hardware

### Lavpasfilter

Der benyttes et lavpasfilter for at undgå aliasering. Dette kaldes derfor for et antialiaseringsfilter. I dette projekt arbejdes med et aktivt 2. ordens lavpasfilter, som består af et pasbånd og et stopbånd.

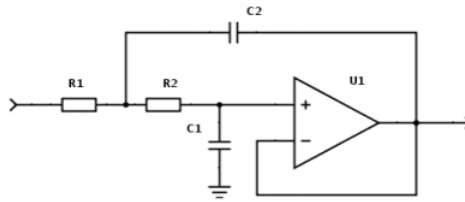


Figur 3.1: Gain respons lavpas

Pasbåndet lader lave frekvenser passere igennem med ingen eller uvæsentlig dæmpning, og stopbåndet dæmper høje frekvenser væsentligt. Kurvens udvikling ses på bodeplot med frekvensen i rad/s ud af x-aksen og forstærkning i dB op ad y-aksen.

Knækfrekvensen er overgangen mellem pas- og stopbånd. Med andre ord så er knækfrekvensen, hvor indgangssignalet er dæmpet med 3 dB.

I projektet designs filtret med en knækfrekvens på 50 Hz. Operationsforstærkeren er af typen OP27. Kondensatoren C2 er givet til 680 nF og endvidere  $R1 = R2$ .



Figur 3.2: Unity 2. ordens sallen-key lavpas konfiguration

Til at bestemme komponentværdier er der taget udgangspunkt i knækfrekvensen:

$$f_c = \frac{1}{2\pi\sqrt{C1 \cdot C2 \cdot R1 \cdot R2}} = \frac{1}{2\pi\sqrt{C1 \cdot C2 \cdot R^2}}$$

Herudfra bestemmes R1 og R2:

$$\text{solve}(50 = \frac{1}{2\pi\sqrt{10^{-6} \cdot (680 \cdot 10^{-9}) \cdot R^2}}, R)$$

$$R = 3860 \, \Omega \approx 3,9 \, k\Omega$$

C1 bestemmes til 1  $\mu F$  Overføringsfunktionen:

$$T_v(s) = \frac{\frac{1}{R1 \cdot C1 \cdot R2 \cdot C2}}{s^2 + s(\frac{1}{R2 \cdot C1} + \frac{1}{R1 \cdot C1}) + \frac{1}{R1 \cdot C1 \cdot R2 \cdot C2}}$$

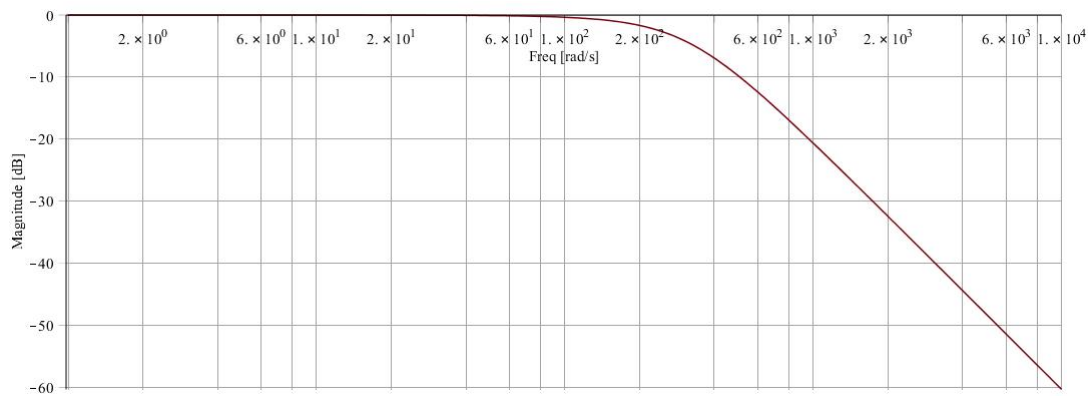
$$T_v(s) = \frac{\frac{1}{(3900(1 \cdot 10^{-6}) \cdot 3900(680 \cdot 10^{-9}))}}{(s^2 + (s(\frac{1}{3900(1 \cdot 10^{-6})} + \frac{1}{3900(1 \cdot 10^{-6})})) + \frac{1}{(3900(1 \cdot 10^{-6}) \cdot 3900(680 \cdot 10^{-9}))})}$$

$$T_v(s) = \frac{1}{\frac{25857}{25000000000}s^2 + \frac{663}{125000}s + 1}$$

$$T_v(s) = \frac{96685,6170476}{s^2 + 512,820512821 \cdot s + 96685,6170476}$$

Optegner bodeplot vha. værktøj i Maple:

$$sys := TransferFunction\left(\frac{96685,6170476}{s^2 + 512,82 \cdot s + 96685,6170476}\right) :$$



Figur 3.3: Bodeplot lavpasfilter

Bodeplottet bekræfter, at det er et lavpas filter. Der aflæses en knækfrekvens ved -3db til 269 rad/s  $\approx$  42,81Hz. Den beregnede knækfrekvens er blevet beregnet til 49,48 Hz. (Evt udregning i bilag) Dette er en relativ lille afvigelse.

### Forstærker

## 3.3 Software

### Trelagsmodellen

Softwaren er opbygget af trelagsmodellen.

# Produktet 4

---

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

---

Tekst	Tekst	Tekst	Tekst.
-------	-------	-------	--------

Tekst	Tekst	Tekst	Tekst.
-------	-------	-------	--------

Tekst	Tekst	Tekst	Tekst.
-------	-------	-------	--------

Tekst	Tekst	Tekst	Tekst.
-------	-------	-------	--------

## Formål

Formålet med dokumentet.





# Accepttest 5

---

Version	Dato	Ansvarlig	Beskrivelse
0.1	30/9-15	HR, JL, LB og SV	Første udkast til accepttest
1.0	8/10-15	Alle	Rettelser efter review
2.0	4/11	HR, SV	Tilrettelser efter rettelser fra vejledere

## Formål

Formålet med dokumentet.

## 5.1 Funktionelle krav

### 5.1.1 Accepttest af use cases

#### Use case 1

<b>Use case under test</b>		UC1: Log ind	
<b>Scenarie</b>		Hovedscenarie	
<b>Forudsætning</b>		Systemet er operationelt	
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Indtast ID "Bruger1"	Det indtastede ID vises på Log ind-GUI		
2. Indtast tilhørende password "1234"	Det indtastede password vises i GUI		
3. Tryk på "Log ind"-knappen	Der bliver logget ind		

*Tabel 5.2: Accepttest af Use Case 1.*

#### Use case 1 - undtagelse pkt. 3a

<b>Use case under test</b>		UC1: Log ind	
<b>Scenarie</b>		Undtagelse 3.a	
<b>Forudsætning</b>		Systemet er operationelt	
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Tryk på <i>Log ind</i> -knappen	Systemet gør opmærksom på fejl, og beder om ny indtastning af ID samt password		

*Tabel 5.3: Accepttest af Use Case 1 - undtagelse.*

**Use case 2**

<b>Use case under test</b>		UC2: Mål blodtryk	
<b>Scenarie</b>		Hovedscenarie	
<b>Prækondition</b>		UC1 er succesfuldt kørt. VPN forbindelse skal være oprettet, når der testes på IHA.	
<b>Handling</b>		<b>Forventet resultat</b>	<b>Resultat</b>
1. Indtast CPR-nummer "1212923434"		Det indtastede CPR-nummer vises i "Diagnostik" GUI	
2. Tryk på " <i>Hent patientoplysninger</i> " - knappen		"Diagnostik" GUI vises med patients CPR-nummer	
3. Nulpunktsjustering foretages		Nulpunktsjustering vises 4. Se efter måling på graf	Målingen kan ses kontinuert på grafen

*Tabel 5.4: Accepttest af Use case 2*

**Use case 2 - undtagelse pkt. 2a**

<b>Use case under test</b> UC2: Mål blodtryk			
<b>Scenarie</b>		Undtagelse 2.a	
<b>Prækondition</b>		UC1 er succesfuldt kørt. VPN forbindelse skal være oprettet, når der testes på IHA.	
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Tryk på " <i>Hent patientoplysninger</i> " - knappen	Systemet gør opmærksom på fejl, og beder om ny indtastning af CPR-nummer		

*Tabel 5.5: Accepttest af Use case 2 - undtagelse 2.a*

**Use case 2 - undtagelse pkt. 3a**

<b>Use case under test</b> UC2: Mål blodtryk			
<b>Scenarie</b>		Undtagelse 3.a	
<b>Prækondition</b>		UC1 er succesfuldt kørt. VPN forbindelse skal være oprettet, når der testes på IHA.	
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Tryk på " <i>Ny nulpunktsjustering</i> "-knappen	Systemet foretager ny nulpunktsjustering		

*Tabel 5.6: Accepttest af Use case 2 - undtagelse 3.a*

**Use case 2 - undtagelse pkt. 4a**

<b>Use case under test</b>	UC2: Mål blodtryk		
<b>Scenarie</b>	Undtagelse 4.a		
<b>Prækondition</b>	UC1 er succesfuldt kørt. VPN forbindelse skal være oprettet, når der testes på IHA.		
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1a. Konstruer højt diastolisk tryk (>140)	"Diagnostik"GUI får et rødt skær og der alarmeres med lyd		
1b. Konstruer lavt diastolisk tryk (<100)	"Diagnostik"GUI får et rødt skær og der alarmeres med lyd		
1c. Konstruer højt systolisk tryk (>90)	"Diagnostik"GUI får et rødt skær og der alarmeres med lyd		
1d. Konstruer lavt systolisk tryk (<60)	"Diagnostik"GUI får et rødt skær og der alarmeres med lyd		
2. Tryk på " <i>Lydløs</i> "	Lyden forsvinder i tre minutter		
3. Normaliser blodtrykket	Alarmen stopper, alarmlyden forsvinder og brugergrænsefladen får et grønt skær		

*Tabel 5.7: Accepttest af Use case 2 - undtagelse 4.a*

**Use case 3**

<b>Use case under test</b>	UC3: Filtrer signal		
<b>Scenarie</b>	Hovedscenarie		
<b>Prækondition</b>	UC2 er kørt succesfuldt		
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Tryk på "Fra" - radiobutton	Systemet slår filteret fra, og viser dette i radiobutton på GUI		
2. Tryk på "Til" - radiobutton	Systemet slår filteret til, og viser dette i radiobutton		

*Tabel 5.8: Accepttest af Use case 3***Use case 3 - undtagelse pkt. 2a**

<b>Use case under test</b>	UC3: Gem data		
<b>Scenarie</b>	Undtagelse 2a		
<b>Prækondition</b>	UC1 og UC2 er gennemført		
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Tryk på "Gem data" -knappen	Sysetemet giver beskeden "Data ikke gemt - tekniker er tilkaldt"		
2. Tryk "OK"	UC3 starter fra pkt. 1		

*Tabel 5.9: Accepttest af Use case 3 - undtagelse 2.a*



**Use case 4**

<b>Use case under test</b>		UC4: Gem data	
<b>Scenarie</b>		Hovedscenarie	
<b>Forudsætning</b>		UC2 er gennemført.	
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Tryk på " <i>Gem data</i> " -knappen	Systemet giver besked den " <i>Data gemt</i> "		
2. Tjek i databasen, om de korrekte data er gemt	De korrekte data er gemt i databasen		
3. Systemet giver besked den " <i>Data gemt</i> "	Pop-up meddelelsen om at data er gemt kommer frem		

*Tabel 5.10: Accepttest af Use case 4*

**Use case 5**

<b>Use case under test</b>	UC5: Kalibrer system		
<b>Scenarie</b>	Hovedscenarie		
<b>Forudsætning</b>	Tekniker er tilkaldt		
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Påtryk systemet tre kendte tryk	Reponserne aflæses på GUI		
2. Noter afvigelserne på de kendte tryk	Afvigelserne noteres		
3. Juster systemets software i forhold til kalibreringen	Justering foretages		

*Tabel 5.11: Accepttest af Use case 4*

**Use case 5 - undtagelse 3a**

<b>Use case under test</b>			
UC5: Kalibrer system			
<b>Scenarie</b>		Undtagelse 3a.	
<b>Forudsætning</b>		Tekniker er tilkaldt	
<b>Handling</b>	<b>Forventet resultat</b>	<b>Resultat</b>	<b>Godkendt</b>
1. Påtryk systemet tre kendte tryk	Reponserne aflæses på GUI		
2. Noter afvigelserne på de kendte tryk	Ingen afvigelser forefindes		
3. UC5 afsluttes	UC5 afsluttes		

*Tabel 5.12: Accepttest af Use case 5 - undtagelse pkt. 3a*

## 5.2 Ikke-funktionelle krav

Krav nr.	Krav	Test	Forventet	Resultat	Godkendt
1	Programmet skal programmeres i C#, Visual Studio	Åbn programmet	Det ses i programmet om det er programmeret i C#		
2	Systemet bør kunne angive pulsen via en lyd ved hvert pulsslag ved ... Hz	Pulsen indlæses i systemet, og frekvensen måles	Pulsen angives af en lyd med ... Hz		
3	Blodtrykket skal kunne gemmes i database	Det tjekkes, om det korrekte data er gemt i en database	Det korrekte data er gemt i databasen		
4	Blodtrykket skal måles inden for 10 mmHg præcision	Det tjekkes at systemet måler blodtrykket inden for den angivne værdi	Blodtrykket er målt inden for 10 mmHg præcision		
5	Forstærkeren skal kunne forstærke det elektriske signal	Et signal sendes igennem forstærkeren	Signalet forstærkes		
6	Lavpasfilteret skal kunne filtrere støj fra det elektriske signal	Et signal sendes igennem lavpasfilteret	Signalet filtreres og støjen fjernes		
7	Programmet skal indeholde en "Log ind"-knap	Det ses i <i>Log ind</i> -GUI, om programmet indeholder en "Log ind"-knap	Programmet indeholder en "Log ind"-knap		

8	Programmet skal indeholde en " <i>Hent patientoplysninger</i> "-knap	Det ses i <i>Diagnostik</i> -GUI, om programmet indeholder en " <i>Hent patientoplysninger</i> "-knap	Programmet indeholder en " <i>Hent patientoplysninger</i> "-knap		
9	Programmet skal indeholde en " <i>Ny nulpunktsjustering</i> "-knap	Det ses i <i>Diagnostik</i> -GUI, om programmet indeholder en " <i>Ny nulpunktsjustering</i> "-knap	Programmet indeholder en " <i>Ny nulpunktsjustering</i> "-knap		
10	Programmet skal indeholde en " <i>Lydløs</i> "-knap	Det ses i <i>Diagnostik</i> -GUI, om programmet indeholder en " <i>Lydløs</i> "-knap	Programmet indeholder en " <i>Lydløs</i> "-knap		
11	Programmet skal indeholde en " <i>Gem data</i> "-knap	Det ses i <i>Gem data</i> -GUI, om programmet indeholder en " <i>Gem data</i> "-knap	Programmet indeholder en " <i>Gem data</i> "-knap		
12	Programmet skal indeholde en " <i>Til</i> "-radiobutton	Det ses i <i>Diagnostik</i> -GUI, om programmet indeholder en " <i>Til</i> "-radiobutton	Programmet indeholder en " <i>Til</i> "-radiobutton		

13	Programmet skal indeholde en " <i>Fra</i> "-radiobutton	Det ses i <i>Diagnostik</i> -GUI, om programmet indeholder en " <i>Fra</i> "-radiobutton	Programmet indeholder en " <i>Fra</i> "-radiobutton		
14	Systemet bør kunne køre fejlfrit i et år	Kan ikke testes			
15	Systemet bør have en MTTR på højst 24 timer	Kan ikke testes			
16	Systemet skal kontinuert vise en grafisk afbildning af blodtrykket, hvor tryk er op af y-aksen og tiden er på x-aksen i intervaller af 6 sekunder	Det ses på <i>Diagnostik</i> -GUI om denne indeholder en grafisk afbildning, hvor tryk er op af y-aksen og tid er op af x-aksen	<i>Diagnostik</i> -GUI indeholder en grafisk afbildning med de korrekte værdier op af y- og x-aksen		
17	Softwareen bør være opbygget af trelagsmodellen	Det ses i programmet, om dette er opbygget af trelagsmodellen	Programmet er opbygget af trelagsmodellen		

Tabel 5.13: Ikke-funktionelle krav



## Bilag 1: Samarbejdsaftale

### Faglige aftaler

- Vi forventer at få lavet et projekt, vi kan stå inde for.
- Vi har en ambition om en over middel præstation.

### Aftaler om gruppens samarbejde

- Alle gruppemedlemmer er aktivt deltagende.
- Vi overholder indbyrdes aftaler.
- Vi arbejder effektivt og viser respekt for andre gruppemedlemmer.
- Alle aftaler indskrives i en fælles kalender, hvor det er eget ansvar at være opdateret.
- Det er eget ansvar at give besked, hvis man er forhindret i at møde til den aftalte tid.
- Der skal være plads til, at gruppemedlemmerne kan have fritidsinteresser.
- Der vil blive uddelegeret hjemmeopgaver, og disse skal laves til den aftalte tid. Hvis man ikke har haft tiden, skal dette meddeles hurtigst muligt til de resterende i gruppen.
- Vi planlægger arbejdstiden inkl. pause. Ingen sjov og surf i arbejdstiden.
- Vi forventer at kunne mødes mindst en gang om ugen.
- Vi forventer at alle gruppemedlemmer kan deltage i vejledermøderne en gang om ugen.
- Vi forventer, at det respekteres, at et gruppemedlem gerne vil være lidt i baggrunden, hvis personen måtte have en dårlig dag.
- Vi forventer at dette er et forum, hvor vi kan vende problemer mellem gruppemedlemmer åbent og derved ikke sidder med problemerne selv.
- Vi har tillid til, at de personer, der har ansvaret for en opgave, har styr på det.
- Vi er indstillet på at kunne tage imod både ris og ros.
- Der er plads til pauser – også individuelle – under gruppens arbejde.
- Gruppen fører en fælles logbog, der opdateres efter dagens arbejde.

### Sanktioner



- Overholder et medlem ikke samarbejdsaftalen, vil gruppen ved enstemmighed kunne ekskludere gruppemedlemmet.

**Bilag 2: Tidsplan**

Version	Dato	Ansvarlig	Beskrivelse
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.

TIDSPLAN	uge 37	uge 38	uge 39	uge 40	uge 41	uge 42	uge 43	uge 44	uge 45	uge 46	uge 47	uge 48	uge 49	uge 50	uge 51	...	uge 4
planlæg projektet	×																
vidensindsamling	×	×															
projektformulering	×	×															
ks + at		×	×	×													
ks-/at-deadline				×													
systemarkitektur			×	×	×		×										
ret sys.ark.								×	×								
sys.ark.-deadline									×								
hw-/sw-iteration*								×	×	×	×						
produkt-deadline														×			
accepttest														×			
rapportskrivning											×	×		×			
korrektur														×	×		
rapport-deadline															×		
off. deadline															×		
forbered fremvis.																	×
fremvisning																	×

\*: Opdelt i teams

Sort: Travl uge

Lilla: Lettere uge

Blå: Normal uge

Brunrød: Deadline

**Bilag 3: Hardware versioner**

Version	Dato	Ansvarlig	Beskrivelse
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.

**Bilag 4: Software versioner**

Version	Dato	Ansvarlig	Beskrivelse
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.
Tekst	Tekst	Tekst	Tekst.

## **Bilag 5: Logbog**

Logbogen findes på vedlagte cd-rom.

## **Bilag 6: Mødereferater**

Mødereferater findes på vedlagte cd-rom.

## **Bilag 7: Datasheet NI-6009 DAQ**