



AARHUS
UNIVERSITET
INGENIØRHØJSKOLEN

Projektrapport

Semesterprojekt 3

Blodtryksmålersystem

ST3PRJ3-03

Gruppe 5

Sundhedsteknologi

Århus Universitet, IHA

Vejleder: Samuel Alberg Thyresøe

Dato: 16/12 2015

Mette Østergård Knudsen, 201404501

Ida Mark Skovbjerg, 201404669

Line Skov Larsen, 201405838

Brian Hansen, 201310502

Mohamed Hussein Mohamed, 201370525

Khaled Edwan, 200800899

Resumé

Formålet med dette projekt er at udvikle et system, som kan tilsluttes et væskefyldt kateter og vise en blodtryksskive på en computerskærm. I klinisk praksis er der behov for kontinuert at vise patientens blodtryk og herfra aflæse systolen og diastolen. Denne patientovervågning sker primært på intensive afdelinger og operationsstuer. Blodtrykket indikerer patientens helbredstilstand og giver herved det sundhedsfaglige personale en sikkerhed omkring patientens tilstand. Dokumentationen for projektet er struktureret efter ASE-modellen, systemet er udviklet ud fra V-modellen, og software samt hardware processen er udarbejdet gennem en iterativ agil udvikling med vandfaldsmodellen som udgangspunkt.

I dette projekt er der valgt en løsning, hvor blodtrykket præsenteres grafisk og værdien for systole, diastole samt middeltryk vises. Der er også udarbejdet to databaser, hvor hhv. patients data og personalets data ligger gemt. I databasen med patientdata gemmes patientens blodtryk løbende. Der er blevet udviklet et system, som kan præsentere et blodtrykssignal, og alarmere hvis dette overskrider de angivende grænseværdier. Blodtrykket er varierende hos forskellige patientgrupper, og disse grænseværdier vil derfor kunne reguleres op og ned. Det kan i dette projekt konkluderes, at vi har udarbejdet et hardware og et software produkt, som ved interaktion kan indhente data og præsentere disse på en brugergrænseflade. Det er lykkedes os at få et blodtrykssignal vist, dog har det tidsmæssigt ikke været muligt, at implementere EKG og ressourcemæssigt har det ikke været muligt at implementere iltmætning.

Abstract

The purpose of this project is to develop a system that can be connected to a liquid-filled catheter and display a blood pressure curve on a computer screen. In clinical practice, it is necessary to continuously display the patient blood pressure and hereby read the systolic and diastolic value. This monitoring of patient occurs in intensive care and operating rooms. Blood pressure is an indicator of the patient's state of health and gives the health professionals evidence about the patient's state of health.

The ASE-model structures the documentation for the project and the system has been developed by the V-model. The software and hardware process is developed through an iterative agile process where the waterfall-model has been a starting point.

In this project we have chosen a solution where blood pressure is presented graphically and where the value for systolic, diastolic blood pressure and mean pressure are shown. Additionally two databases have been made where respectively patient data and healthcare professional data are stored. The patient database stores patient blood pressure regularly. The system has been developed, which can provide a blood pressure signal and an alarm starts if the blood pressure exceeds the limit for normal blood pressure. Blood pressure varies in different patient groups and therefore it is possible to change the limits.

For the project we have succeeded in developing a product prototype where an interaction between hardware and software can obtain data and present those on a computer screen.

It has been possible for us to show a blood pressure signal, however from a time consuming point of view it has not been possible to implement EKG and due to lack of sufficient resources we have not been able to implement oxygen saturation.

Indhold

Indhold	ii
1 Indledning	1
2 Projektformulering og afgrænsning	2
2.1 Oversigt over projektdeltagere og hovedansvarsområder	3
3 Hjertet	4
3.1 Blodtryk	4
3.2 Direkte invasiv måling af blodtryk	6
4 Systembeskrivelse	8
5 Krav	10
5.1 Aktørbeskrivelse	11
5.2 Use case beskrivelse	12
5.3 Ikke-funktionelle krav	13
6 Projektbeskrivelse	15
6.1 Projektgennemførelse	15
6.2 Metode	18
6.3 Specifikation og analyse	19
6.4 Arkitektur	20
6.5 Design, implementering og test	25
6.6 Resultater og diskussion	33
6.7 Udviklingsværktøjer	38
6.8 Opnåede erfaringer	38
6.9 Fremtidigt arbejde	39
7 Konklusion	41
Litteratur	42
7.1 Referencer	42

Kapitel 1

Indledning

I dette projekt er der arbejdet med design og udvikling af et blodtryksmålesystem. Ideen bag vores system blev udtænkt og udarbejdet med viden om en blodtryksmåler på en operationsstue på Herning Sygehus. Vores opstillede krav samt design til systemet, blev derfor lavet ud fra de oplysninger vi fik af en sygeplejerske¹ på Herning Sygehus. Der var et ønske om at lave et blodtryksmålesystem, som er så virkelighedsnært som muligt. Det design og den funktionalitet der blev efterstræbt i dette projekt, var at præsentere et blodtrykssignal, hvor det systoliske samt det diastoliske blodtryk kan bestemmes. Desuden få præsenteret et EKG-signal, hvorfra pulsen kan bestemmes, og præsentere iltmætningen i blodet for patienten. Vi ønsker at få vist grafer og værdier på en brugergrænseflade, som lever op til de standarder, der ses på operationsstuer i dag.

Baggrunden for dette projekt er, at man i klinisk praksis har behov for kontinuert at observere patientens blodtryk. Det er i særdeleshed på intensiv afdelinger og operationsstuer, at man ønsker at lave direkte invasiv blodtryksmåling. Ved invasiv blodtryksmåling kan man monitorere patientens fysiske tilstand under eksempelvis en operation. Blodtrykket kan give en indikation om blødninger, smerter og hvor dybt patienten sover under narkosen. Denne overvågning giver det sundhedsfaglige personale en ekstra tryghed og sikkerhed om patientens tilstand under operation.

Denne projektrapport vil give en kort beskrivelse af vores samlede system samt krav hertil. Der vil være en udspecificeret projektbeskrivelse, som vil give et indblik i vores vej frem til det endelige resultat. Den vil også beskrive de til- og fravalg vi har måttet tage, for at nå frem til det system, der kunne realiseres og præsenteres. Der har i den forbindelse været nogle ideer, som ikke har været realistiske og tidsmæssigt mulige at opfylde, og vil derfor være beskrevet i fremtidigt arbejde. Slutteligt vil der være en samlet konklusion på projektarbejdet.

¹Anæstesisygeplejerske Charlotte Høj, Herning Sygehus

Kapitel 2

Projektformulering og afgrænsning

I dette projekt udarbejdes et produkt, der kan måle blodtrykket for en patient og vise dette løbende på en graf. Desuden skal produktet kunne bestemme det diastoliske og systoliske tryk ud fra blodtrykskurven, ud fra disse trykværdier bestemmes middeltrykket. Ud fra systolisk og diastolisk tryk skal produktet kunne vise om en eller begge har overskredet en grænseværdi og signalere dette til det sundhedsfaglige personale. Produktet skal kunne filtrere det indhentede signal med et digitalt filter, sådan at signalet kan vises filtreret og ufiltreret.

Dette betyder at produktet skal kunne:

- Indhente et blodtrykssignal igennem hardwaren, hvilken består af:
 - Væskefyldt kateter
 - Transducer
 - Forstærker
 - Lavpas filter
 - Spændingsforsyning
 - NI-DAQ
- Vise blodtrykskurven kontinuert
- Vise blodtrykskurven både filtreret og ufiltreret
- Bestemme systolisk tryk
- Bestemme diastolisk tryk
- Bestemme middeltryk
- Starte alarm, hvis systolisk/diastolisk tryk har overskredet en grænseværdi
- Gemme blodtryksdata i en database

Løsningen til dette produkt har for vores vedkommende været et software og et hardware produkt, hvilke benyttes sammen til at indhente, behandle og vise blodtrykssignalet. Hardwaren bruges til at indhente og behandle blodtrykssignalet, sådan at dette signal kan sendes til softwaren. Softwaren skal kunne viderebehandle signalet, sådan at det indlæste signal kan præsenteres efter de opsatte krav. Softwaren skal have en brugergrænseflade, som bruges til at præsentere

blodtrykssignalet for det sundhedsfaglige personale.

Ligeledes er det et ønske at produktet skal indhente EKG-signalet og ilt saturationen fra patienten. Disse data skal også kunne vises på en brugergrænseflade. Ud fra EKG-signalet bestemmes pulsen, altså hjertefrekvensen, for patienten, pulsen kan dog også bestemmes ved at finde antallet af systoler pr. minut, denne værdi præsenteres som talværdi på brugergrænsefladen. Ud fra ilt saturationen kan iltmætningen for blodet bestemmes, hvilken ligeledes præsenteres på brugergrænsefladen.

2.1 Oversigt over projektdeltagere og hovedansvarsområder

I dette projekt er det valgt at dele projektets deltagere op i to grupper med hver deres hovedansvarsområder. Den ene gruppe har beskæftiget sig med hardware og den anden gruppe med software. Derudover blev der i starten af forløbet valgt en projektleder, hvis opgaver bestod i at lave dagsorden for hvert møde, sørge for at deadlines blev overholdt, træffe endelige beslutninger og have et overblik over projektet. En procesleder blev ligeledes valgt, hvis opgaver bestod i opgavestyring, godt arbejdsmiljø i gruppen og planlægning af projektførelsen.

Herunder ses de forskellige gruppemedlemmers hovedområde:

Hardware	Software
Brian Hansen	Ida Mark Skovbjerg
Mohamed Hussein Mohamed	Mette Østergård Knudsen
Khaled Edwan	Line Skov Larsen

Gruppens medlemmer med fulde navn og initialer ses herunder::

Fulde navn	Initialer
Ida Mark Skovbjerg	IMS
Line Skov Larsen	LL
Mette Østergård Knudsen	MK
Brian Hansen	BH
Mohamed Hussein Mohamed	MM
Khaled Edwan	KE

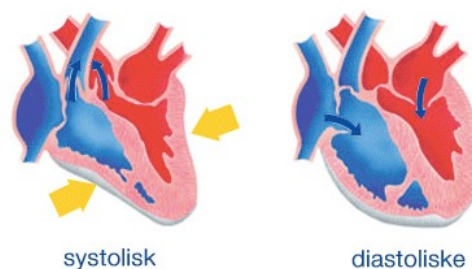
Kapitel 3

Hjertet

Hjertet er en muskel, der fungerer som en pumpe i kroppen. Hjertet pumper det iltede blod (arterielt blod) ud til resten af kroppen gennem arterie- og vene systemet. Når blodet passerer ud af hjertet igennem arterierne, sker der et tryk mod arterievæggene, dette tryk kaldes blodtrykket.

Hjertet opdeles i to adskilte hjertehalvdele, som fungerer som pumpe. Den venstre, som pumper blodet rundt i organismen og den højre, som pumper blodet gennem lungerne. De to hjertehalvdele er sammensat af atrier og ventrikler. Blodet fra organismen der føres tilbage til hjertet, er det venøse blod, som føres til højre atrium gennem øvre og nedre hulvene. Mellem højre atrium og højre ventrikel sidder trikuspidalklappen, som forhindrer blodet i at løbe tilbage til atriet under hjertets sammentrækning. Højre ventrikel udløber i lungepulsåren, som deles ud i hver lunge. Mellem ventriklen og lungepulsåren er pulmonalklappen, som forhindrer blodet i tilbageløb. I lungerne afgiver blodet CO_2 og optager ilt, denne diffusion af disse gasser sker i lungernes alveoler. Det nu iltrige blod returneres gennem lungevenerne til venstre atrium. Fra venstre atrium til venstre ventrikel sidder mitralklappen. Fra venstre ventrikel pumpes blodet ud i aorta gennem aortaklappen, herfra pumpes blodet rundt i resten af organismen. [2]

Systolisk og diastolisk blodtryk



Figur 3.1: Hjertet ved systolisk og diastolisk blodtryk.

3.1 Blodtryk

Blodtrykket ses når hjertets venstre hjerteventrikel trækker sig sammen, og der som resultat heraf, bliver presset blod ud i arterierne. Trykket vil altså være størst i arterierne, da arterierne er de blodåre som fører væk fra hjertet. Det laveste blodtryk vil være at se i venerne, da det

er de blodårer, som fører blodet tilbage til hjertet. Blodtrykket vises som to værdier: systolen, som er hjertets sammentrækningsfase og diastolen som er hjertets afslapningsfase.

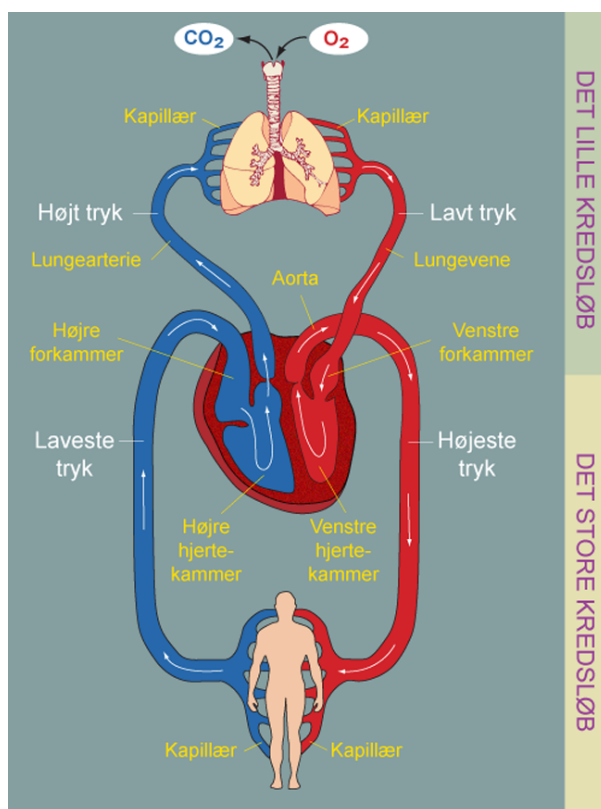
Systolen angives som den kraft, der skabes når der kommer pres på karvæggene i arterierne. Dette sker i hjertets sammentrækningsfase, hvor blodet pumpes ud i arterierne. Systolen i et normalt blodtryk i hvile vil ligge i intervallet 100-140 mmHg og ved forhøjet blodtryk er værdien for systolen over 140 mmHg.

Diastolen angiver hjertets hvilefase og ses mellem to sammentrækninger. I denne fase fyldes ventriklerne med blod. Den normale værdi for diastolen ligger i intervallet 60-90 mmHg. Hvis den overstiger 90 mmHg anses blodtrykket for at være forhøjet. Udover systolen og diastolen kan middeltrykket også angives. Dette udregnes ved $Middeltryk = \frac{2}{3} \cdot Diastole + \frac{1}{3} \cdot Systole$ [3]

Hypertension defineres som forhøjet systole, forhøjet diastole eller både forhøjet systole og diastole. Hypertension belaster hjertet, og kan medføre en øget risiko for udviklingen af apopleksi, arteriosklerose, hjerteinsufficiens og nyreskader. [5]

Hvis værdien for systolen kommer under 100 mmHg og/eller diastolen kommer under 60 mmHg, vil der være tale om et for lavt blodtryk, hypotension. Lavt blodtryk behøver dog ikke at være alvorligt, men kan være helt normalt for f.eks. høje personer. Men i forbindelse med sygdom, traumer eller medicinoverdosering er hypotension en alvorlig tilstand. Hypotension ses bl.a. ved diabetes, traumer, hjertesvigt og Addisons sygdom. [6]

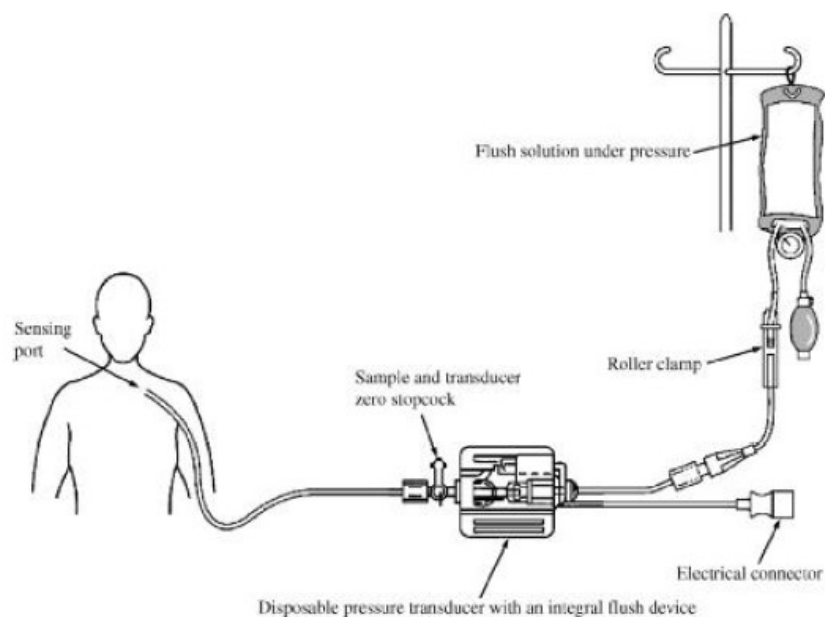
Anatomien bag blodtrykket og blodtryksændring, omfatter blodets kredsløb, hjertet og lymfesystemet. Blodkredsløbet er et langt netværk af arterier og vener, som har til opgave at føre iltet blod ud til hver celle i kroppen via kapillærerne, samt opfange affaldsprodukter fra celler og væv.[5] I lungekredsløbet udskilles CO₂ og blodet iltes. Det er hjertet der pumper blodet ud i dette netværk, så blodet hele tiden er i bevægelse. Arterierne transporterer blodet væk fra hjertet, og venernes opgave er at transportere blodet mod hjertet.



Figur 3.2: Det lille og det store kredsløb. Motoren der driver det hele er hjertet. [1]

3.2 Direkte invasiv måling af blodtryk

Den mest almindelig metode til klinisk at måle blodtrykket, er ved at koble det vaskulære tryk til en extravaskulær sensor via et væskefyldt kateter. Klinisk er dette system påsat patientens a. Radialis via en arteriekanyle. Kateterslangen er forbundet til en trevejs stophane og videre til tryksensoren. Kateter-sensor systemet er fyldt med en saltvandsopløsning fra væskesøjlen.



Figur 3.3: Opstilling af direkte invasiv blodtryksmåling.

Som det ses på figuren er sansningsporten indgangen til patients arterie. Kateteret er koblet til trevejs stophanen og videre til sensoren. Trevejs stophanen har tre mulige funktioner, 1: Der kan lukkes for blodtilførslen så der er åbent for atmosfærisk luft og transduceren, dette giver værdien til nulpunktsjustering. 2: Der kan åbnes for atmosfærisk tryk ind til blodåren, dette kan bruges hvis patienten skal modtage medicin. Her indsættes medicin og ved at åbne for trykket fra væskesøjlen, vil medicinen flyde med blodet ind i blodåren. 3: Der åbnes for blodtilførsel ind i transduceren, dette giver blodtrykket. Blodtrykket transmitteres via kateter væskesøjlen ind i sensoren og til sidst i membranen. Det er her blodtrykket bliver målt.

Kapitel 4

Systembeskrivelse

Det er blevet besluttet ud fra projektformuleringen at udvikle et blodtryksmålesystem som en prototype med perspektivering til fremtidigt arbejde. Blodtryksmålesystemet ønskes ideelt at kunne måle blodtryk, EKG og iltmætning for en patient. Ud fra blodtrykket findes systolisk og diastolisk blodtryk, baseret på disse værdier kan middeltrykket bestemmes. Systole og diastole bestemmes ved at finde den maksimale værdi (systole) og den minimale værdi (diastole) for blodtrykskurven. Middeltrykket kan herefter bestemmes med formlen:

$$Middeltryk = \frac{1}{3} \cdot Systole + \frac{2}{3} \cdot Diastole. [3]$$

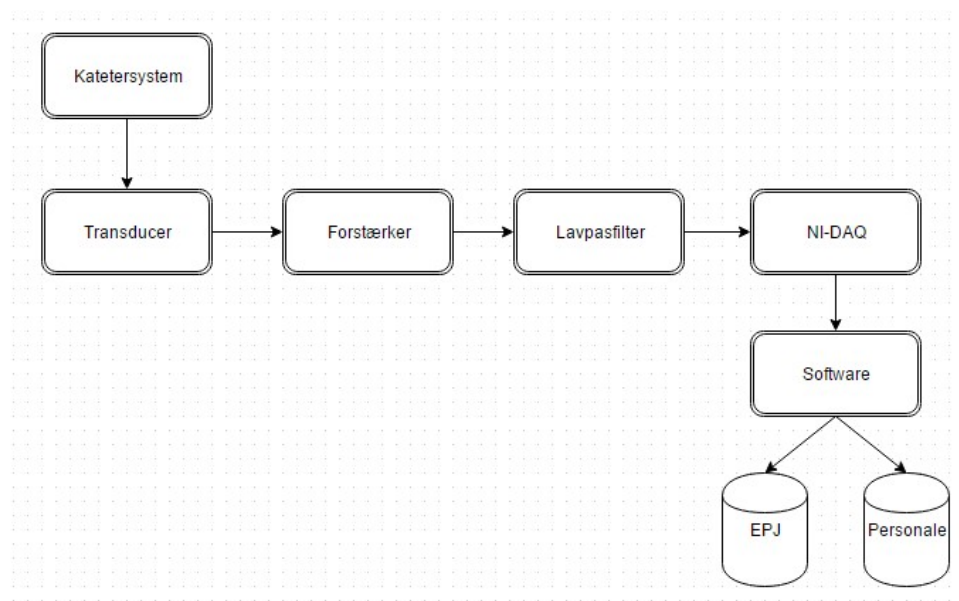
Ud fra EKG-signalet kan pulsen bestemmes, dette gøres ved at bestemme antallet af R-takker på et minut. Desuden kan pulsen bestemmes ud fra blodtrykket, da pulsen her er antallet af systoliske værdier på et minut. Iltmætningen er mængden af ilt i blodet. For at kunne bestemme denne værdi skal et eksternt produkt benyttes. Dette produkt skal ved hjælp af en infrarød sensor bestemme iltmætningen i blodet, dette bliver ikke implementeret i denne prototype.

Forudsætningerne for brug af systemet er, at det skal køres på en computer, samt overholde de opstillede krav. Systemet er opbygget af en hardwaredel og en softwaredel.

Hardwaredelen består af et aktivt 2. Ordens Butterworth Sallen-Key lavpasfilter, samt en forstærker. Forstærkeren modtager et signal fra den udleverede transducer, dette signal forstærkes op. Signalet sendes videre til lavpasfilteret, hvor alle frekvenser over 50 Hz bliver dæmpet. Herefter sendes signalet ind i dataopsamlingsenheden NI-DAQ.

Software delen består af et windows forms program udviklet i C# .NET, programmet skal kunne præsentere data indhentet fra DAQ'en. Ligeledes skal programmet kunne analysere og filtrere data fra DAQ'en, samt gemme disse i en "EPJ-database. Programmet skal også kunne hente login-oplysninger fra en "personale-database.

Det er valgt at have to databaser for at simulere, at der er en adskillelse af personaledata og patientdata.



Figur 4.1: Skitse af opbygning af systemet.

Kapitel 5

Krav

Hvilke krav der stilles til produktet, udarbejdes i en kravspecifikation. Denne kravspecifikation består af en række Use cases. Disse Use cases beskriver interaktionen mellem aktørerne og systemet. Use cases har til formål at specificere, hvilke krav der, stilles til produktet. Kravene opstilles ud fra, hvad kunden ønsker, samt hvad leverandøren finder muligt at realisere.

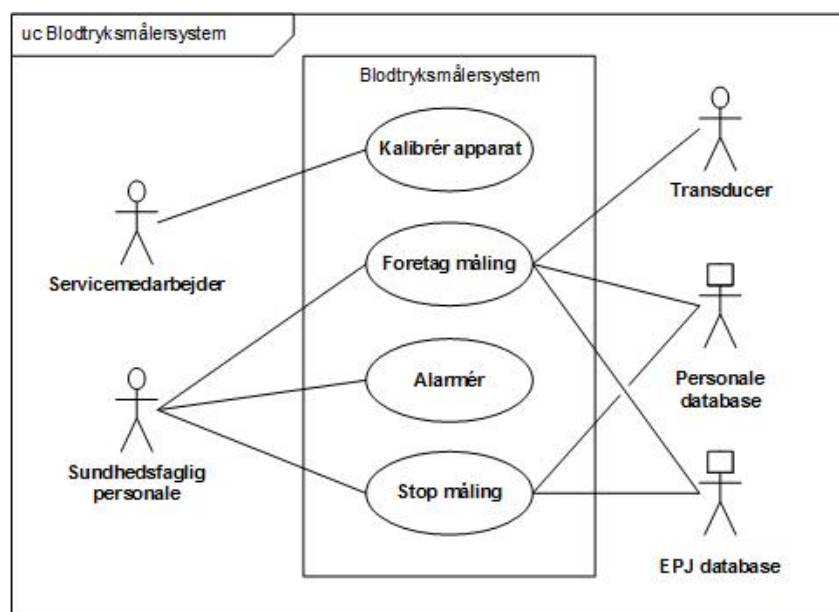
Der er nogle obligatoriske krav til produktet, der skal opfyldes. Disse krav er bl.a., at systemet skal kunne kalibrere det indhentede blodtrykssignal og foretage en nulpunkts justering. Desuden skal blodtrykssignalet vises kontinuerligt, hvilket betyder, at signalet skal vises grafisk, samtidigt med at det bliver hentet/indsendt data fra hardwaren. Disse behandlede data skal desuden gemmes i en database. Et af de obligatoriske krav er desuden, at der skal være et digitalt filter, der kan slås til og fra når programmet kører. Dette filter skal sørge for, at signalet haves i to forskellige modes; et diagnose mode, hvor signalet er rå med alle udsving, og et monitor mode, hvor signalet er filtreret og afrundet.

Foruden disse obligatoriske krav er det valgt, at det skal være muligt at starte og stoppe målingen. Når målingen er startet skal det være muligt at justere grænseværdierne for systolen og diastolen, så grænseværdierne kan tilpasses patienten. Når signalets værdier overskrider en af grænseværdierne, skal en alarmering begynde. Denne alarm skal kunne udsættes i et minut, dog kun alarmens lyd, så det stadig er tydeligt at se, at en grænseværdi er overskredet.

Systemet består af en computer med en programkode, en NI-DAQ, et lavpas filter, en forstærker, en transducer og en In vitro maskine.

Blodtrykket dannes i denne prototype af en In vitro maskine, som leverer et tryk til en tryktransducer, herefter sender tryktransducere signalet videre til hardwaren, hvor signalet bliver forstærket og høje frekvenser bliver dæmpet. Dette signal sendes derefter igennem NI-DAQ ind til softwaren, som derefter behandler og analyserer signalet.

Den fulde beskrivelse af de udarbejdede Use cases (fully dressed Use case skema) findes i dokumentationen under Kravspecifikation.



Figur 5.1: Use Case diagram. Dette diagram viser aktørernes interaktion med systemet.

5.1 Aktørbeskrivelse

Ud fra use case diagrammet ses de seks aktører; *Sundhedsfaglig personale*, *Transducer*, *EKG patient*, *EPJ database*, *Personale database* og *Servicemedarbejder*. Det er disse aktører der interagerer med systemet.

Sundhedsfaglig personale

Det er det sundhedsfaglige personale, der påsætter måleudstyret på patienten. Det sundhedsfaglige personale er desuden den aktør der interagerer med systemet; logger på og foretager en måling. Det sundhedsfaglige personale har dermed adgang til systemet og de viste målinger på brugergrænsefladen.

Transducer

Transduceren agerer som patienten, denne består af en In vitro maskine tilsluttet katetersystemet. Katetersystemet leverer et tryk i mmHg videre til tryktransduceren, som omdanner trykket til et analogt signal. Dette signal sendes videre i hardwaren, som består af en forstærker og et lavpasfilter, som behandler signalet. Signalet behandles i NI-DAQ'en som sendes ind i systemet. Derfor er denne aktør kilden til måleresultaterne for systole, diastole og middeltryk.

EKG-patient

EKG patienten er kilden til EKG-kurven, da der ikke haves en rigtig patient, hvorfra både EKG og blodtryk kan indhentes.

Værdierne for denne aktør hentes fra PhysioBank ATM.

Servicemedarbejder

Servicemedarbejderen står for at foretage kalibreringen. Dette gør Servicemedarbejderen ved at koble systemet til kalibreringssystemet, og indtaste de værdier for tryk (mmHg) og spænding (Volt), som kan måles ved tre forskellige målepunkter på en vandsøjle. Herefter kan en kalibreringsværdi findes af systemet.

EPJ database

I EPJ databasen gemme patient- og måledata. Blodtryksskurvens rådata gemmes som en talliste. For at have nok information om signalet gemmes kalibrerings- og nulpunktsjusteringsværdien for signalet også i EPJ databasen. Patienterne i denne database kobles sammen med en sundhedsfaglig person fra Personale databasen.

Personale database

Det sundhedsfaglige personales login-informationer ligger i Personale databasen.

5.2 Use case beskrivelse

Use case diagrammet viser ligeledes de fire Use cases, der er for systemet: Kalibrér apparat, Foretag måling, Alamér og Stop måling. Disse Use cases er en beskrivelse af, hvad systemet skal kunne, og beskriver de interaktioner, der sker mellem aktørerne og systemet.

Use case 1: Kalibrér apparat

Servicemedarbejderen skal i denne Use case foretage en kalibrering. Dette gøres ved at der er tre målepunkter på en væskesøjle, hvor én af disse målepunkter vælges. Fra målepunktet måles spændingen (volt), med et multimeter eller Analog Discovery, og trykket (mmHg) aflæses fra væskesøjlen. Disse indtastes af Servicemedarbejderen ind i systemet igennem brugergrænsefladen; startskærmen. Herefter beregner systemet en kalibrerings værdi. Denne værdi bruges til at omskrive de indlæste værdier fra DAQ, til en blodtrykskurve i mmHg.

Use case 2: Foretag måling

Denne Use case indhenter data og præsentere disse. For at dette kan gøres, skal det Sundhedsfaglige personale først logge på systemet. Dette gør aktøren ved at indtaste brugernavn og adgangskode på brugergrænsefladen; startskærmen. Når den sundhedsfaglige er logget på, henter systemet de tilknyttede patienter, hvorefter det Sundhedsfaglige personale kan vælge den ønskede patient. Når Patienten er blevet valgt startes hovedskærmen.

For at sikre at graferne, der hentes ligger rigtigt på graferne, skal det Sundhedsfaglige personale nulpunktsjustere systemet, sådan at graferne kommer til at ligge rigtigt på akserne. Dette gøres ved, at der tages højde for det atmosfæriske tryk.

Herefter kan den Sundhedsfaglige starte målingen. Når dette gøres, henter systemet blodtryks-signalet og EKG-signalet. Disse bruges af systemet til at beregne systole, diastole, middeltryk og puls. Systemet viser graferne kontinuert på hver sin graf og systole, diastole, middeltryk og puls vises som talværdier. Samtidig gemmer systemet løbende disse data i EPJ databasen.

Det vil være muligt for det Sundhedsfaglige personale at slå det digitale filter fra og til. Filteret er pr. default slået til fra start.

Det er også muligt for det Sundhedsfaglige personale at justere grænseværdierne for systolen og diastolen, dette gøres for at indstille grænseværdierne individuelt for hver patient.

Use case 3: Alarmér

Alarmen startes af systemet, når en af grænseværdierne for systole eller diastole overskrides. Dette gør at knappen, der benyttes til at udsætte alarmen, blinker skiftevis mellem at have rød og hvid baggrund, og der starter en alarm lyd. Når alarmen er igangsat kan endnu en grænseværdi overskrides, hvis dette sker, er der umiddelbart ikke nogen ændring. Alarmlyden fortsætter fra tidligere, og knappen, til udsættelse af alarmen, fortsætter med at blinke.

Det vil her være muligt for det sundhedsfaglige personale af udsætte alarmen. Dette gøres ved at trykke på "Udsæt alarm"knappen, som illustreres med et alarm billede, hvorefter systemet stopper alarmens lyd i et minut. Efter dette minut starter alarmens lyd igen. "Udsæt alarm"knappen fortsætter med at blinke, indtil forholdene er normaliseret, altså indtil grænseværdien ikke længere er overskredet.

Use case 4: Stop måling

Det Sundhedsfaglige personale skal også kunne stoppe målingen. Dette gøres ved, at det sundhedsfaglige personale interagerer med hovedskærmen. Herefter vil det være muligt for det Sundhedsfaglige personale at logge ud, og et nyt scenarie kan påbegyndes.

5.3 Ikke-funktionelle krav

Ikke funktionelle krav er opsat efter FURPS+ metoden. Kravene er herefter blevet prioriteret efter MoSCoW. De udspecificerede ikke-funktionelle krav kan findes i Dokumentationens under Kravspecifikationen.

FURPS+

Functionality

Funktionalitet er det brugeren ønsker sig. Dette omfatter også sikkerhedsrelaterede behov. Dette er krav til hvad programmet skal kunne af funktionalitet, f.eks. at programmet skal gemme data løbende i en database.

Usability

Hvor effektiv er produktet, set fra forbrugers side, dette er det aspekt brugervenlighed ser på. Er produktet nemt at bruge. Hvordan bruges produktet; er der nogen brugergrænseflader. Det er herunder kravene til hvilke knapper der skal være på brugergrænsefladerne, og dermed også hvilke brugergrænseflader der skal være.

Reliability

Pålidelighed tager sig af aspekter som, hvor længe er det maksimalt at systemet må være nede. Er der fejl der kan forudses. Hvor præcist kan resultaterne vises. Er produktet let at vedligeholde; kan delene i produktet skiftes let.

Performance

Præstationen for produktet, handler om hvor hurtigt produktet er om at starte op og hvor hurtigt svartiden er. Hvor stor må svartiden maksimalt være. Så det er altså herunder, at det

er beskrevet, hvor lang tid der går fra der er trykket på en knap, til at systemet svarer.

Suportability

Produktets support fortæller, om det er muligt at teste på produktet, om det er muligt at udvide produktet, installere og konfigurere produktet. Desuden om produktet er kompatibelt. Det er herunder programmets opbygning beskrives

+

Kunden kan have nogle yderligere behov, disse yderligere behov beskrives under +. Hvilke begrænsninger er der ved designet. Er der nogle krav for brugergrænsefladerne. Er der nogle fysiske eller implementeringskrav. Er der dele der kan genbruges, herunder hele systemer eller dele af dem. Det er bl.a herunder at kravene til computeren, der benyttes, beskrives. [8]

MoSCoW

Must

De krav der bliver markeret som et must er de krav som produktet skal have. Altså det produktet skal have/indeholde.

Should

De krav der markeres som et should-krav, er de krav til produktet der burde være med. Altså er det hvad produktet bør indeholde

Could

Kravene kan også markeres som could. Disse krav er de krav der kunne være gode at have med, men som ikke bliver prioriteret højt. Så det er kun, hvis man kan nå at få det med, at de skal være der.

Would/Won't

Det er ikke alle krav, der skal være gældende for produktet, disse krav markeres som would/won't. Det er altså disse krav som man ikke tager med eller de krav som ville være sjove at have med, men ikke har en betydning for produktet, men er en tilføjelse eller udvidelse. Det er disse krav der danner rammen for fremtidigt og videregående arbejde med projektet.

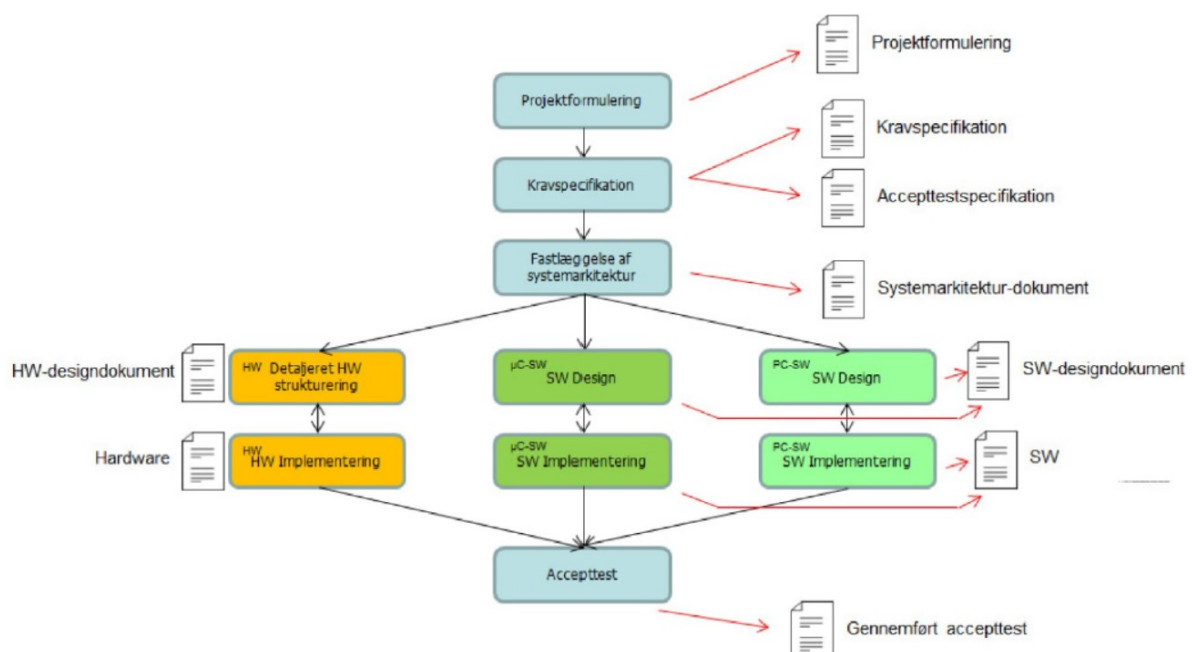
Kapitel 6

Projektbeskrivelse

6.1 Projektgennemførelse

ASE-modellen

Til udviklingen af dette produkt er der primært benyttet ASE-modellen. Modellen ses herunder. ASE-modellen er en udviklingsmodel, der er udarbejdet af Aarhus Ingeniørhøjskole. Denne model er en mellemvægtig semi-iterativ udviklingsproces, hvilken drives ud fra Use cases. Projektudviklerne fastlægger først en projektformulering, hvorfra en kravspecifikation kan udarbejdes, hvilken indeholder kravene fra kunden. Systemarkitekturen fastlægges ud fra kravspecifikationen, for derefter at designe og implementere de enkelte hardware og software dele hver for sig i iterationer. Til sidst fører dette til en accepttest, hvilken gennemføres, så der opstår en enighed mellem projektudviklerne og kunden.



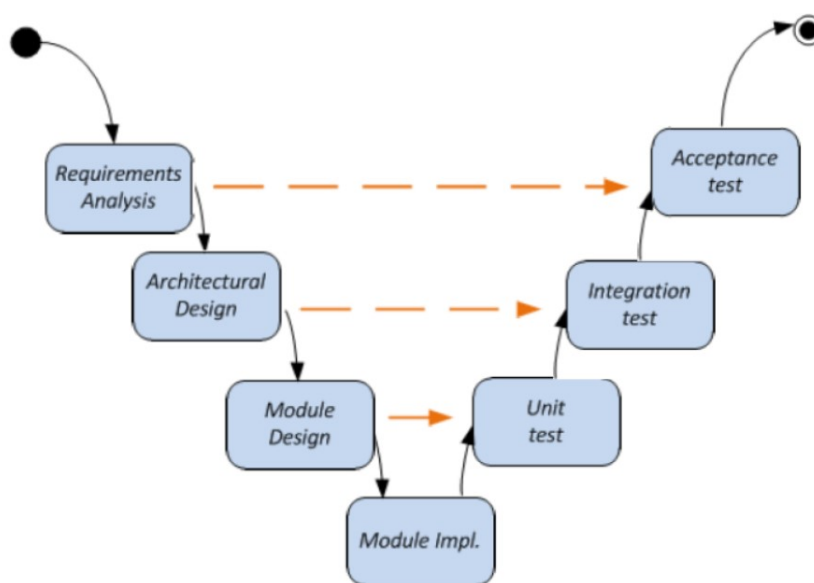
Figur 6.1: ASE-modellen.

Kravspecifikationen bliver specificeret som en række Use cases ud fra problemformulerin-

gen. Use cases benyttes til at beskrive aktørernes interaktion med systemet. Sammen med de ikke-funktionelle krav opnås der med Use casene et overblik over hvilke krav der stilles til systemets funktionalitet. Systemets design bestemmes igennem hardware og software design og implementering, herefter kan en accepttest udarbejdes. Ved udførelsen af accepttesten tjekkes det at kravene er opnået.

V-modellen

V-modellen er en faseopdelt udviklingsmodel. Denne model beskriver udviklingsfaserne og testfaserne i et projekt sideløbende. V-modellen er blevet benyttet sideløbende med ASE-modellen. Specifikationen af test foregår parallelt med udviklingen af selve systemet i V-modellen. Fordelen ved denne model er at testene sker på forskellige niveauer, hvilket sikrer at udviklede delsystemer virker som ønsket. Det vigtige her er at en fase er færdiggjort, inden den næste fase påbegyndes.

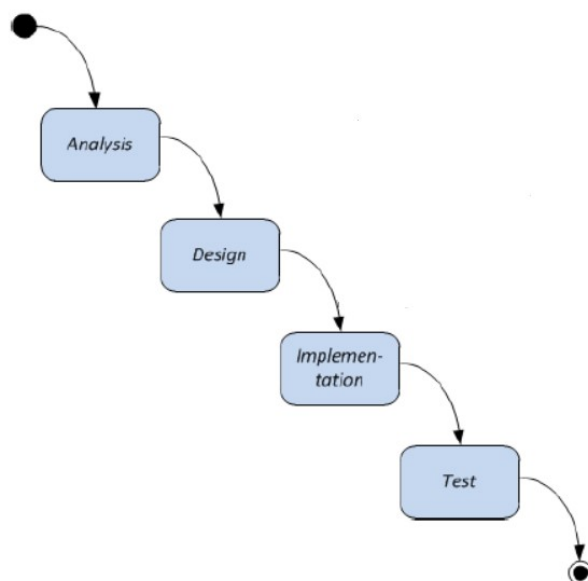


Figur 6.2: V-modellen.

Ud fra modellen ses det at den første fase er udviklingen af kravspecifikationen. Her udarbejdes der en tilhørende accepttest, denne test gør det muligt at tjekke om systemet til sidst lever op til de opstillede krav. Den næste fase er herefter systemarkitektur, her udvikles den tilhørende test, denne test undersøger integrationen mellem de implementerede moduler. De sidste to fase af udviklingsfasen er design og implementering af systemets enkelte moduler, her udføres enhedstesten løbende af de implementerede moduler.

Vandfaldsmodellen

Vandfaldsmodellen er en model der bruges til udvikling af software og hardware, hvor software- og hardwareudviklingen betragtes som værende konstant nedadløbende. Her kan udviklingen af et modul i en ny fase først påbegyndes når den foranliggende fase er afsluttet. Denne model er i projektet benyttet til udviklingen af software- og hardwarearkitekturen.



Figur 6.3: Vandfalds modellen.

Disse tre modeller; ASE-modellen, V-modellen og vandfalds-modellen arbejder alle i en kronologisk logisk rækkefølge. Modellerne er benyttet i et omfang, hvor de har kunnet supplere hinanden. ASE-modellen giver det store overblik, mens V-modellen sikrer udarbejdelsen af de nødvendige tests. Vandfalds-modellen er den proces, hvilken der er arbejdet ud fra i alle projektets facetter.

Projektstyring

Arbejdsfordeling

Som nævnt tidligere valgte man fra projektets start at opdele gruppen i to hold, henholdsvis et softwarehold og et hardwarehold.

Ændringer i projektstyring

Under projektforløbet blev det valgt at ændre denne rollefordeling, idet den generelle arbejdsfordeling samt arbejdsindsats i projektgruppen ikke fungerede optimalt. Det blev valgt, at kun et gruppemedlem skulle varetage rollen som projektleder og procesleder, men at opgaverne herunder kunne uddelegeres hvis nødvendigt. Denne leder blev i fællesskab valgt og baggrunden for dette, var et ønske om en leder, som både havde viden og indflydelse på hardware og software. Denne leder skulle gribe ind, hvis gruppearbejdet og deltagernes indsats ikke opfyldte den forventningsafstemning, som står skrevet i samarbejdskontrakten.

Scrum

Det blev i øvrigt besluttet at benytte SCRUM som hjælp til styring af projektet. Der blev udnævnt en tovholder, som skulle holde møder i løbet af projektets udarbejdelse, samt styre opgaverne. Man valgte at bruge Pivotal Tracker [9] til at styre og prioritere opgaver. Der var en overordnet tidsplan fra start, denne tidsplan består af en række deadlines, som kan findes i dokumentationen.

GitHub

Det blev besluttet at benytte GitHub til deling af informationer og dokumentation i forbindelse med udarbejdelse af rapporten. Det blev vurderet, at det var vigtigt, at der var en central informationsudveksling, så begge grupper kunne følge med i, hvor langt de var i de forskellige processer.

Iterativ udviklingsproces

Udviklingsprocessen på både software- og hardwareholdet blev meget iterativ, det betød at der var mange forskellige versioner af både hardware og software. Man valgte denne metode da store dele af projektet var et teknologistudie, så det var nødvendigt at lave mange iterationer løbende. Desuden var det vigtigt, at de to hold kunne omstille sig hurtigt, og bruge ny viden uden at skulle igennem en langsommelig design proces på ny.

6.2 Metode

[11] For at beskrive den overordnede systemarkitektur og det detaljerede design for produktet, er der blevet benyttet SysML og UML. SysML er et grafisk modelleringssprog, hvilket kan hjælpe til at forstå og udvikle komplekse systemer. SysML udspringer af UML, dog benyttes SysML i højere grad ved systemer, der både indeholder software og hardware. UML benyttes til at beskrive softwaresystemets struktur og forløb. [7]

SysML's struktur-, funktionalitets- og adfærdsdiagrammer er i projektet blevet benyttet til at specificere og dokumentere systemet og dets komponenter. Der anvendes to strukturdiagrammer; block definition diagram (BDD) og internal block definition diagram (IBD) for at vise systemstrukturen. Systemarkitekturen bliver delt op i blokke. BDD'et bruges til at vise forholdet mellem de forskellige blokke, deres komposition og forholdet mellem logiske- og fysiske enheder i systemet. IBD'et benyttes til at vise den forbindelse der er imellem de forskellige blokke der ses ud fra BDD'et. Dermed ses den vej signalet går igennem de forskellige enheder og hvordan disse enheder er koblet sammen.

For at kunne beskrive produktets funktionalitet, er der blevet udarbejdet et Use case diagram. Dette diagram beskriver, hvordan systemet interagerer med de udefrakommende enheder (f.eks. aktører) for at løse et antal opsatte opgaver. Disse opgaver er beskrevet i forskellige Use cases. Domænemodellen beskriver det overordnede systemdomæne. Systemdomænet beskrives ud fra de konceptuelle klasser, hvilke findes ud fra Use cases. Domænemodellen definerer de klasser, der skal være i softwaren og de interaktioner der er imellem disse, dette gøres med klassediagrammer og adfærdsdiagrammer.

Et af adfærdsdiagrammerne er et sekvensdiagram, denne viser de handlinger, der er mellem systemets dele (parts), hvilke er delt op i sekvenser.

For at beskrive overgangen fra de beskrevne Use cases til software, er applikationsmodellen benyttet. Applikationsmodellen tager udgangspunkt i Use cases og domænemodellen, hvor Use cases bruges til at identificere, hvad systemet skal kunne.

En applikationsmodel med kontrolklasser, domæneklasser og boundaryklasser kan opbygges. Kontrolklasserne beskriver hvordan data behandles mellem domæne- og boundaryklasserne. Domæneklasserne indeholder funktionaliteten som det pågældende softwaremodul benytter, det er herfra klasserne, der skal bruges i softwaren identificeres. Boundaryklasserne har den funktion

at beskrive, hvordan systemet kommunikerer med omverdenen og omvendt. Ved at benytte applikationsmodellen til design af softwarens opbygning, gøres det lettere at udvikle et produkt med lav kobling samt høj samhørighed.

Systemets softwarestruktur og forløb beskrives ved at benytte UML klassediagrammer. Klassediagrammet definerer relationerne mellem de forskellige klasser. Derfor dannes der et hurtigt overblik over sammenhængen i systemet ved at benytte klassediagrammet.

6.3 Arkitektur

Hardware

Systemets hardwarearkitektur beskrives på baggrund af systembeskrivelsen og kravspecifikationen. Ud fra relevante diagrammer, som BDD og IBD, kan hardwarens arkitektur beskrives. Samtlige diagrammer og modeller kan findes i Projektdokumentationen under Arkitektur og design.

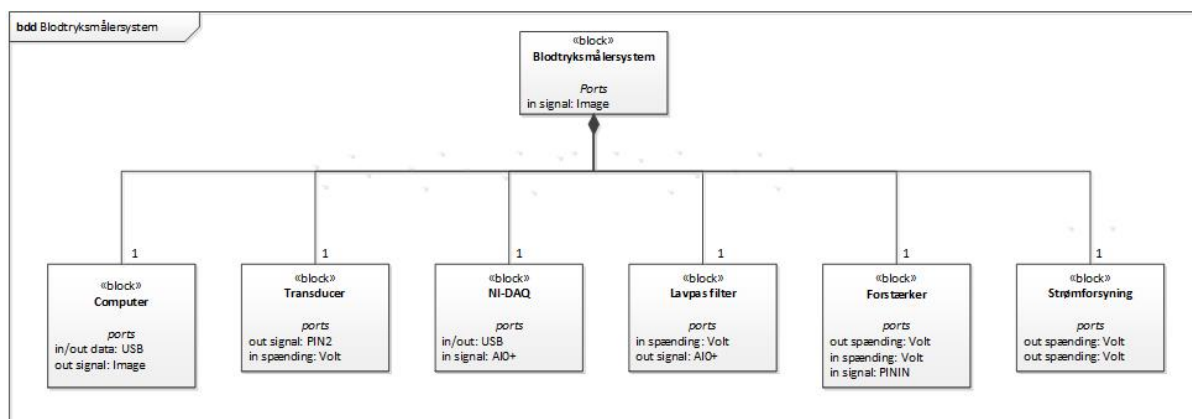
Problem identifikation

Først skulle der identificeres hvad produktet skal kunne, og hvordan det skulle virke. Der er i første omgang opstillet krav til hardwaren ud fra kravspecifikationen. De første krav til hardwaren var at, hardware delen skulle bestå af en operationsforstærker og et lavpasfilter. Der blev i første gang arbejdet med et lavpasfilter, som ikke var aktivt. Denne ide blev kasseret, da der efter senere overvejelser blev opdaget at lavpasfilteret skulle være et anden ordens filter af typen sallen-key. Herved blev kredsløbet for hardwaren bygget op efter en operationsforstærker og et anden ordens lavpas filter af typen sallen-key. Da filteret er aktivt, er der tilsat en spændingsforsyning i form af to 9V batterier.

Hardwaren er herved yderligere specificeret igennem udformningen af diagrammerne BDD og IBD. Diagrammerne beskriver hvilke blokke blodtryksmålesystemet er bygget op af, og signalets vej igennem blokkene.

BDD - block definition diagram

Ud fra use casen og kravene til hardware omkring blodtryksmåler systemet, er der identificeret hvilke blokke blodtryksmålesystemet består af. Der er identificeret og beskrevet hvordan hver blok i blodtryksmålesystemet snakker sammen, og er indført på BDD'et som stregerne imellem blokken. Der er angivet hvilke signaler blokkene sender ud imellem hinanden.

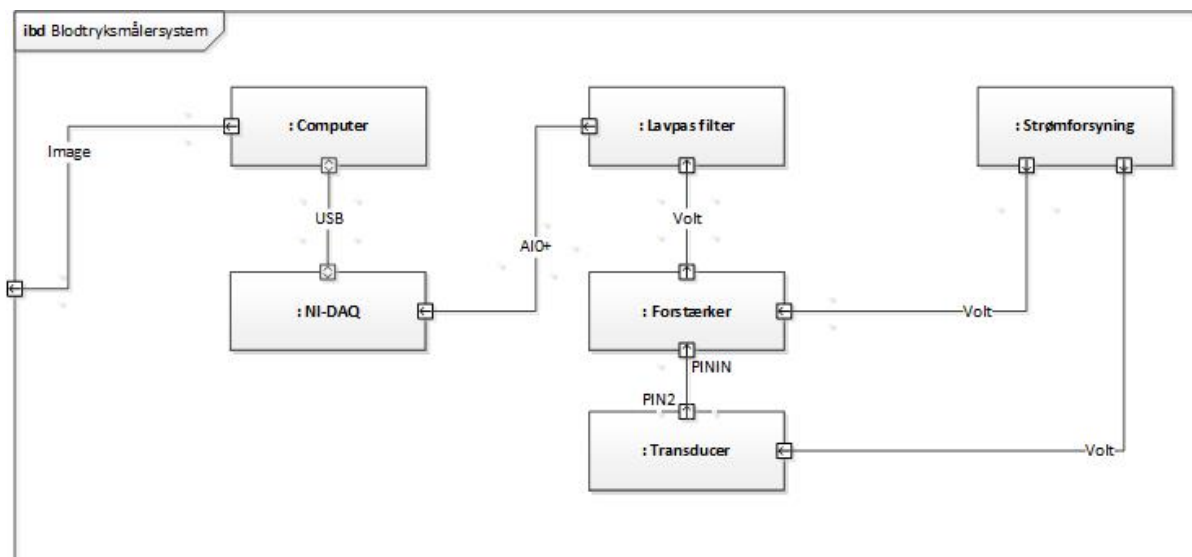


Figur 6.4: BDD for blodtryksmålesystemet.

Ud fra diagrammet ses det at alle blokkene er koblet til blodtryksmålesystemet, som er selve softwaren for blodtryksmåleren. Kompositionen i diagrammet viser at blodtryksmålesystemet består af en computer, en transducer, en NI-DAQ, et lavpasfilter, en forstærker og en spændingsforsyning. Disse blokke til sammen sørger for at forstærke et målt signal, derefter filtreres de høje frekvenser fra, hvor efter signalet bliver omformet fra et analog signal, til et digitalt signal i NI-DAQ, som blodtryksmåleren indlæser blodtryksdata fra.

IBD - internal block definition diagram

Ud fra block definition diagram kan forbindelserne mellem blokkene identificeres. Dette føre til at et IBD diagram kan opstilles. Et IBD bruges til at bestemme signalerne mellem blokkene, og identificere hvilke signaler der sendes ind og ud af hver blok. Det er ud fra IBD der bestemmes hvilke signal typer, som blodtryksmålesystemet kan integrerer med.



Figur 6.5: IBD diagram for blodtryksmålesystemet.

Software

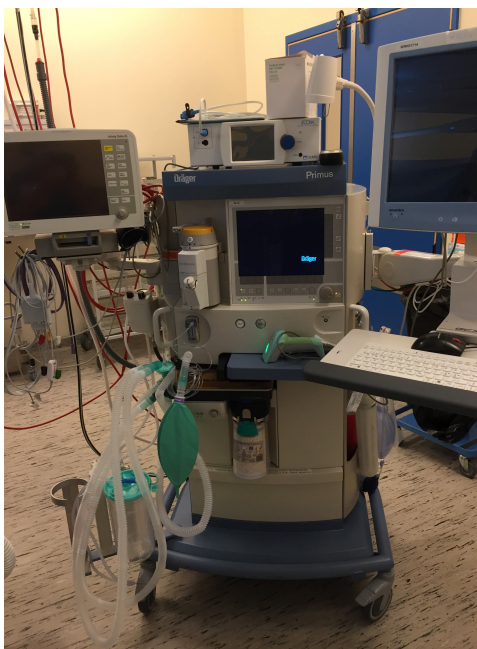
Systemets softwarearkitektur beskrives på baggrund systembeskrivelsen og kravspecifikationen. Ud fra relevante diagrammer og modeller kan software beskrives hvorfra arkitekturen for softwaren opnås. Samtlige diagrammer og modeller kan findes i Projektdokumentationen under Arkitektur og design.

Problemidentifikation

Den første fase i produkt udviklingen bestod i, at få klarlagt hvad produktet skulle bruges til og hvilke funktioner dette skulle have. Den første ide som blev udtænkt var en blodtryksmåler, som skulle benyttes til diabetes patienter. Denne ide førte til første udkast af startskærmen. På startskærmen kunne man vælge, hvor på patienten målingen skulle foretages. Her kunne man altså vælge at blodtryksmålingen skulle foretages invasivt i a. Brakialis samt underekstremiteterne og sammenligne disse to. Dette ville give en diagnosticering om hvor slemt diabetes er hos patienten. Dette kan klarlægges idet blodomløbet til underekstremiteterne hos diabetikere er dårlig, da der ses åreforkalkning hos diabetikere. Dog blev denne ide kasseret, idet der med et lavt blodtryk vil være dårlig sårheling og større risiko for infektion.

Ideen til produktet blev udtænkt efter samtale med anæstesisygeplejerske Charlotte Høj fra operationsafdelingen på Herning Sygehus. På billedet ses anæstesisygeplejerskens arbejdsfelt, dette består af (fra højre) EPJ system, anæstesi apparat og monitor.

Vi ønskede at vores produkt skulle opbygges med udgangspunkt i et brugsscenarie og efterleve dette system, dog kun EPJ og monitor, og de hertil tilhørende funktioner. EPJ systemet på Herning sygehus benyttes både til patientjournal og personale login. Det er her sygeplejersken logger på med et unikt bruger-ID og der vil til dette være knyttet de patienter, som hun skal tilgå den pågældende dag. Sygeplejersken vil derfor kunne se relevante data for patienten. Derudover vil det målte blodtryk også gemmes i EPJ systemet. Monitoren er den brugergrænseflade, som grafisk præsenterer blodtrykket, EKG og iltmætning samt tilhørende værdier. På monitoren er der en start- og slukknapp og en timer, som starter når blodtryksmålingen sættes i gang. Derudover er der indbyggede grænseværdier for sysolen og diastolen, som justeres afhængigt af patientens tilstand.

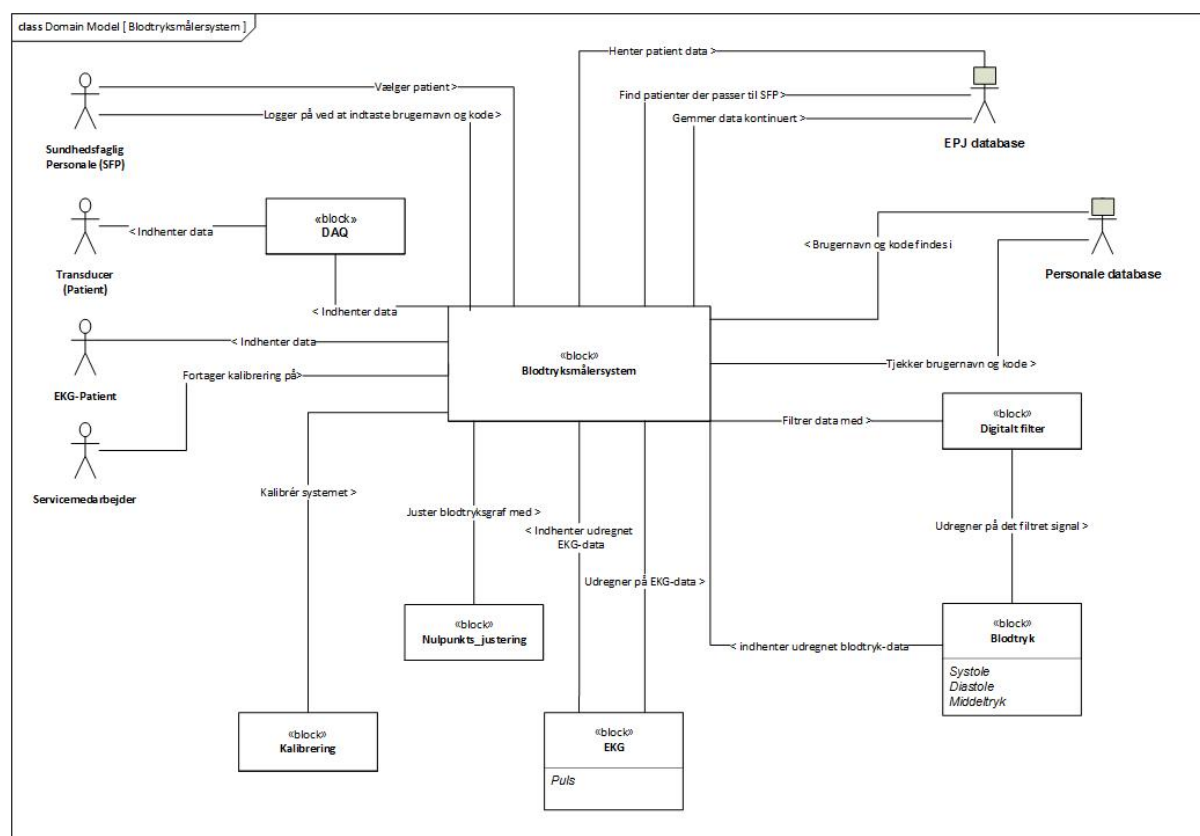


Figur 6.6: Anæstesisygeplejerskens arbejdsfelt.

Ud fra denne idé kunne Use cases udarbejdes, disse er beskrevet under Krav, hvor hovedscenariet for brugen af blodtryksmåleren beskrives.

Domænemodel

Ud fra disse Use cases kunne de klasser som systemet skulle bestå af identificeres. Disse klasser identificeres ved at bestemme de konceptuelle klasser, som indeholder den information som systemet skal holde styr på. De konceptuelle klasser indføres i domænemodellen som klasser. Det er i domænemodellen hvor problemet i forhold til hvad der skal holdes styr på i softwaren kan bestemmes.



Figur 6.7: Domænemodel for blodtryksmålersystemet.

Ud fra denne model ses det sundhedsfaglige personales interaktion med systemet, samt hvilke handlinger der igangsættes af denne interaktion. Det sundhedsfaglige personale udfører en handling, der medfører, at en række processer starter i systemet. Disse processer sørger for at hente data fra transduceren og EKG patient, samt sørger for at starte beregningen af værdierne for puls, systole, diastole og middeltryk. Efter beregningerne viser systemet disse værdier på brugergrænsefladen, samt sørger for at disse data bliver gemt i EPJ database.

Klasseidentifikation

Ud fra domænemodellen kan softwareklasserne altså identificeres. Dette fører til at en applikationsmodel kan opstilles. Applikationsmodellen bruges til at bestemme de interagerende klasser i blodtryksmålersystemet og hvordan disse klasser kan tale sammen. Det er altså også ud fra applikationsmodellen at det kan bestemmes, hvilke klasser der skal ligge i de forskellige lag i trelagsmodellen, da klasserne i applikationsmodellen kan være af typen; domain, boundary og control.

Controlklasserne (kontrolklasserne) er de klasser der udfører Use casene ved at interagere med domain- og boundary klasserne. Boundaryklasserne (grænseklasserne) er de klasser der repræsenterer aktørerne fra Use Casene og disse aktøres grænseflader. Domæinklasserne (domæneklasserne) er de klasser hvori data bliver behandlet og bearbejdet. Domæneklasserne er de klasser der skal ligge i logiklaget. Præsentationslaget og datalaget indeholder begge boundaryklasserne, det skal derfor identificeres hvilke indeholder data og hvilke klasser hvorfra interaktionerne startes.

Metodeidentifikation

Ud fra klasserne identificeret af domænemodellen og applikationsmodellen, kan sekvensdiagrammer laves. Sekvensdiagrammerne er interaktionsdiagrammer og viser derfor hvordan processerne forløber i forhold til hinanden. Der laves et sekvensdiagram for hver Use case. Ud fra sekvensdiagrammerne kan det altså ses hvornår og hvordan de forskellige processer forløber i systemet og interagerer med hinanden. Softwaremetoderne kan derfor identificeres ud fra sekvensdiagrammerne. Ud fra sekvensdiagrammerne kan et klassediagram laves. Dette diagram viser hver metode og attribut i hver klasse.

6.4 Design, implementering og test

Hardware

Design

Fra projektets start var der mange informationer og krav til hardwaren i projektet. Det betød at det ret tidligt blev tydeligt hvilke delkomponenter hardwaren skulle bestå af. Hardwaren skulle bestå af et lavpasfilter og en forstærker del, samt en spændingsforsyning. Udviklingsproces og design af hardwaren var igennem en række iterationer, disse er nærmere beskrevet i dokumentationen, dette afsnit vil fokusere på den endelige dimensionering og det endelige design af hardware-delen.

Lavpasfilter

Det var givet at det var et aktivt 2. Ordens Butterworth Sallen-key filter der skulle være lavpasfilter delen af projektet. Det var fra start givet at der skulle benyttes en OP27 operationsforstærker, ligeledes var en af kondensatorerne givet $C2 = 680\text{nF}$ og $R1 = R2$. Filteret skulle dimensioneres til en cut-off frekvens på 50Hz, med en 20dB/decade dæmpning ved 500Hz. Ud fra disse informationer, samt en kredsløbsteget af filteret kunne vi opstille en overføringsfunktion for kredsløbet.

$$T_v(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{s^2 + \frac{R1 + R2}{R1 \cdot R2 \cdot C2} \cdot s + \frac{1}{R1 \cdot R2 \cdot C1 \cdot C2}} \quad (6.1)$$

Denne overføringsfunktion omskrives til standardformel:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\omega^2}{s^2 + 2\zeta\omega + \omega^2} \quad (6.2)$$

Her isoleres cut-off frekvensen (ω) i overføringsfunktionen for lavpasfilteret:

$$\omega = 2\pi\sqrt{\frac{1}{R1 \cdot R2 \cdot C1 \cdot C2}} \quad (6.3)$$

I denne ligning indsatte vi så de kendte værdier, det er beskrevet filteret er af typen 2. Ordens Sallen-Key filter, derfor ved vi at $C1$ er halvdelen af $C2$. [10]

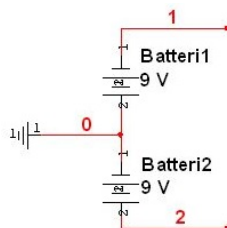
$$50\text{Hz} \cdot 2 \cdot \pi = \sqrt{\frac{1}{R \cdot R \cdot 680 \cdot 10^{-9}\text{F} \cdot 340 \cdot 10^{-9}\text{F}}} \quad (6.4)$$

$$R = 6,62 \cdot 10^3\Omega \quad (6.5)$$

Så vi kan se vores modstande er ca. $6.6k\Omega$, realiseret med to $3.3k\Omega$ modstande i serie i det endelige produkt.

Forstærker INA-114 og valg af spændingsforsyning

Udviklingen af forstærkerdelen til hardwaredelen blev bestemt ved hjælp af databladet for DAQ (NI-DAQ6009), samt databladet for tryk-transducere (TruWave™). DAQ kan maksimalt modtage op til $\pm 10V$, det vil sige vi må ikke overstige peak-to-peak forstærkning på 20V. Før vi kunne beregne forstærkningen skulle vi vælge en spændingsforsyning til kredsløbet, det blev valgt at benytte to 9V-batterier. Disse blev sat op som følger.



Figur 6.8: Spændingsforsyning.

Denne opsætning ville i teorien sikre os 18V peak-to-peak. Man valgte at benytte en INA-114 forstærker til at forstærke signalet fra tryktransducere. I teorien ville man kunne forstærke op til 18V peak-to-peak, men i praksis ville det være urealistisk. Det blev besluttet at forstærke op til 16V peak-to-peak. Undersøgelser for hvorvidt INA-114 havde tilstrækkelige båndbredde blev udregnet som følger. Først bestemte man det maksimale output for tryktransducere i området 0-250mmHg.

$$V_{max} = 250mmHg \cdot 16V \cdot 5\mu V/V/mmHg = 20mV$$

Ud fra dette kan man bestemme hvor meget gain der skal bruges fra forstærkeren.

$$16V = 20 \cdot 10^{-3}V \cdot gain$$

$$gain = 800$$

Man sikrede sig at forstærkeren kunne håndtere forstærkningen, denne del er nærmere beskrevet i dokumentationen. Herefter skulle man beregne hvor stor modstanden på INA-114 skulle dimensioneres til for at få den valgte gain.

$$G = 1 + \frac{50k\Omega}{R_G}$$

$$R_G \approx 62.5\Omega$$

Det blev besluttet at bruge et potentiometer i stedet for en fast modstand til R_G . Så man kunne regulere denne hvis spændingsforsyningen var mere potent end 16V peak-to-peak.

Spændingsregulator

Det blev i den sidste iteration besluttet at benytte en spændingsregulator for at sikre at transducere fik en stabil spændingsforsyning. Tryktransducere er meget følsomme så derfor var det vigtigt at denne får en konstant spændingsforsyning. Det betyder også at den oprindelige skalering ikke er korrekt.

$$V_{max} = 250mmHg \cdot 16V \cdot 5\mu V/V/mmHg = 20mV$$

Gain beregning

$$16V = 6,25 \cdot 10^{-3} VZ \cdot gain$$

$$gain = 2560$$

Produktet af gain og båndbredde er konstant, derfor er det vigtigt båndbredden ligger over knækfrekvensen på 50Hz. Ved gain=1 kan INA-114 levere 1Mhz båndbredde, man kan opstille følgende ligning for beregning af båndbredden.

$$1000000 = 2560 \cdot BW$$

$$BW \approx 390Hz$$

Da 390 Hz er over knækfrekvensen på 50Hz har forstærkeren tilstrækkelig båndbredde. Den nye RG kan beregnes til at skulle være.

$$2560 = 1 + \frac{50k\Omega}{R_G}$$

$$R_G \approx 19,5\Omega$$

Implementering

Implementeringsprocessen har været præget af de mange ændringer, som er foretaget undervejs i hardware designet.

Designet af hardwaren har været igennem flere ændringer undervejs, da bestemmelsen af de rigtige specifikationer til både filtret og operationsforstærkeren ikke har været lige til. Hver gang gruppen har identificeret en fejl i hardware designet som medførte ændring i kredsløbet, skulle implementering af systemet også ændres. De nævnte ændringer er yderligere beskrevet i projektdokumentationen under afsnittet arkitektur og design for hardware. Implementeringsstrategien bestod af to faser. Den første fase var at realisere kredsløbet på et fumlebræt, således at fejl i systemet nemt kan ændres uden store tidsomkostninger.

Når denne fase er gennemført, skulle systemet overføres på et print som skulle designes vha. Multisims Ultiboard. Gruppens medlemmer havde ikke en forudgående erfaringer med at designe print på Multisim, men viljen til at udforske området var tilstede. Der er lagt en del tid på at færdiggøre printet, men undervejs i processen har gruppen valgt at prioritere den første strategi, eftersom tidsforbruget på at designe printet blev uoverkommeligt.

Efter færdiggørelsen af det endegyldige design blev elektronikkredsløbet bygget på et fumlebræt og testen af kredsløbet blev igangsat. Der er anvendt Analog Discovery til simulering af signalet fra transduceren. Dette gøres ved at generere et differentielt signal fra Analog Discovery, hvor W1 og W2 skal tilkobles til de to indgange på INA-114. De to signaler har samme stelpunkt. Udgangs signalet måles vha. en oscilloskop. De to signaler fra henholdsvis W1 og W1 er på 20mV med varieret frekvenser. Det forventet resultat er at de 20mV bliver forstærket og ved frekvens 500Hz skal signalet dæmpes med 40dB på en dekade. Inden accepttesten skulle eksekveres blev kredsløbet overført på et veroboard for at overskueliggøre kredsløbet, da de mange ledninger på fumlebrættet nemt kunne medføre fejl under accepttesten. Denne beslutning var ikke en del af den oprindelige plan, eftersom gruppen havde tiltro til at designe et print til tiden. Resultatet af simuleringen og accepttestens udførelse vil blive fremlagt i afsnittet resultater og diskussion.

Software

Ud fra arkitekturen kan designet af softwaren bestemmes. Det bestemmes hvilke mønstre der skal benyttes og hvordan data skal transporteres igennem koden. Herudover beskrives hvordan nulpunktsjusteringen og kalibreringen foretages.

Kalibrering

Kalibreringen er blevet implementeret sådan at der indtastes en værdi for trykket som vandsøjlen leverer i mmHg og en værdi for spændingen i volt, hvilken kan aflæses fra waveforms eller måles vha. et multimeter. Disse to værdier bliver herefter sendt ned i software klassen Kalibrering, hvor en kalibreringsværdi/omsætningsværdi kan udregnes ved formlen:

$$X \left[\frac{mmHg}{V} \right] = \frac{pressure [mmHg]}{voltage [V]} \quad (6.6)$$

Den værdi der her udregnes benyttes herefter til at omregne den spænding i volt, der indsendes i systemet, til tryk i mmHg.

$$voltage [V] \cdot X \left[\frac{mmHg}{V} \right] = pressure [mmHg] \quad (6.7)$$

Hermed ganges kalibreringsværdien på samtlige værdier i den liste af data der kommer fra DAQ'en (Blodtrykssværdiliste).

Nulpunktsjustering

Nulpunktsjusteringen foregår ved at transduceren er tilsluttet og måler det atmosfæriske tryk, denne værdi skal herefter trækkes fra samtlige værdier for blodtrykket fra DAQ'en. Denne værdi skal derfor trækkes fra inden kalibreringsværdien ganges på. Måden hvormed nulpunktsjusteringen foregår på, er ved at transduceren er tilsluttet og vha. waveforms kan det atmosfæriske tryk aflæses, denne værdi aflæses som en spænding. Denne værdi indtastes på hovedskærmen, hvorefter nulpunktsjusteringen kan startes. Systemet går herefter ind i klassen Nulpunkts_justering, hvor følgende formel bruges til at beregne den værdi der skal benyttes videre:

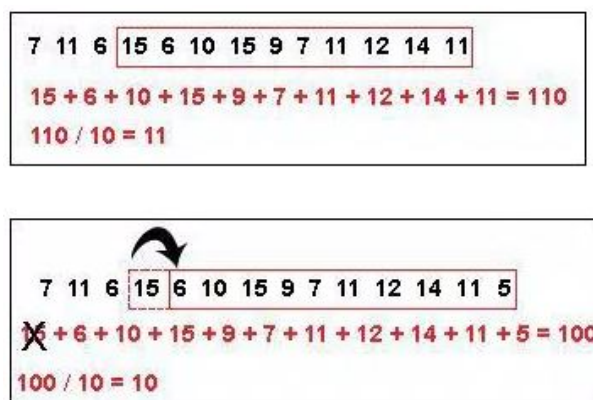
$$V_{zero} [V] = V_{read} [V] - pressure [V] \quad (6.8)$$

Det er denne værdi der herefter bliver kalibreret, for derefter at kunne blive vist som en graf for blodtrykket i mmHg.

Digitalt filter

Det digitale filter der er blevet implementeret i projektet er af typen et glidende middelværdi (moving average) filter. I et glidende middelværdi filter, lægges der et vindue oven på ens talværdi, i dette vindue tager man gennemsnittet af alle værdierne. [?] Vinduet er her betegnelsen for et bestemt tidsinterval.

I projektet er der valgt længden ti på vinduet, det betyder at for hver tiende tal i vores blodtryksliste, bliver der udregnet gennemsnittet af. Herefter rykker vinduet sig til højre og udregner for de næste ti tal i blodtrykslisten. Blodtrykslisten er listen af blodtrykssværdier, der bliver indlæst af DAQ'en.



Figur 6.9: Digital filter ramme. Denne viser hvordan rammen bestemmer gennemsnittet og rykker én hver gang.

Længden af vinduet på ti er valgt, da der bliver vist ti tal fra blodtrykslisten på en graf på brugergrænsefladen. Glidende middelværdifilter kan beskrives med formelen:

$$y[n] = \frac{1}{M} \cdot \sum_{k=0}^M x[n-k], \quad (6.9)$$

$$b_k = \frac{1}{M} \quad (6.10)$$

M er de forgående samples for signalet, n er de nuværende værdier af blodtrykket og k er koefficienterne. Filteret udglatte signalets høje peak værdier, da filteret er et lavpasfilter.

Beregning af systolisk, diastolisk og middeltryks værdi

Algoritmerne for udregning af systole, diastole og middeltryk er alle implementeret i klassen Blodtryk.

Systolen er toppunkterne i blodtryksdataene. Dette kræver dog at man skrive en kort algoritme, så man kan holde styr på hvor langt man er kommet i blodtrykslisten og hvor mange systole værdier der er fundet.

Der er valgt at lave en baseline, som er den grænseværdi der skal overskrides, inden man tjekker efter for toppunkter. Grænseværdien baseline er sat til 100, da den normale værdi for systolen ligger omkring 120 mmHg.

Algoritmen BeregnSystole starter med en for løkke, hvor man har en tæller "i", der starter fra nul og tæller op til længden af blodtrykslisten. Inden i for løkken er der en if løkke, som tjekker om indeks "i" i blodtrykslisten er større end den baseline der er sat. Hvis indeks "i" i blodtrykslisten er større end baseline, vides der at grafen er på vej opad. Herved sættes der en ny baseline, som er lig med værdien for indeks "i" i blodtrykslisten. Der bliver lagt én til værdien baselineoverskredet.

Hvis indeks "i" i blodtrykslisten er mindre end baseline og baselineoverskredet er større end 1, vides der at grafen for blodtrykslisten falder. Når blodtrykslisten falder vides der at den forgående indeks i blodtrykslisten er toppunktet. Der bliver gemt det forgående indeks i blodtrykslisten i en liste der hedder toppunkter. Listen af toppunkterlisten med værdier, er derfor listen af systole værdierne. Baselineoverskredet sættes lig nul, samtidig med at baselinen sættes lig default

værdien, der var 100 mmHg. Algoritmen returnerer en liste af toppunkter.

Diastolen er minimumspunkterne i blodtrykslisten. For at finde minimumspunkterne i listen, er der implementeret en algoritme kaldet *BeregnDiastole*, som finder minimumspunkterne og gemmer dem i en liste.

Algoritmen er opbygget på samme måde som med *BeregnSystole*, med en lille ændring af at algoritmen skal finde minimumspunkter i stedet for toppunkter.

Der sættes en baseline værdi, der er den værdi der skal tjekkes for om indekset i blodtrykslisten er mindre ind. Default værdien for baseline er sat 90 mmHg, for at være sikker på at indekset i blodtrykslisten finder sig neden for denne værdi.

Algoritmen *BeregnDiastole* starter med en for løkke, hvor man har en tæller "i", der starter fra nul og tæller op til længden af blodtrykslisten. Inden i for løkken er der en if løkke, som tjekker om indeks "i" i blodtrykslisten er mindre end den baseline der sat. Hvis indeks "i" i blodtrykslisten er mindre end baseline, vides der at grafen er på vej opad. Herved sættes der en ny baseline, som er lig med værdien for indeks "i" i blodtrykslisten. Der bliver lagt én til værdien baselineoverskredet.

Hvis indeks "i" i blodtrykslisten er større end baseline og baselineoverskredet er større end 1, vides der at grafen for blodtrykslisten stiger. Når blodtrykslisten stiger vides der at det forgående indeks i blodtrykslisten er minimumspunkter. Der bliver gemt det forgående indeks i blodtrykslisten i en liste der hedder minimumspunkter. Listen af minimumspunkter med værdier, er derfor listen af diastole værdierne. Baselineoverskredet sættes lig nul, samtidig med at baselinen sættes lig default værdien, der var 90 mmHg.

Algoritmen til udregning af middeltrykket, udregner middeltrykket med [3] formelen:

$$Middeltryk = \frac{2}{3} \cdot Diastolisketryk + \frac{1}{3} \cdot Systolisketryk \quad (6.11)$$

Algoritmen middeltryk starter med en for løkke med en tæller "i", der starter fra nul og tæller op til og med længden af blodtrykslisten. Inde i for løkken udregnes ligningen for middeltrykket for indeks "i", med systolelisten for indeks "i" og diastolelisten for indeks "i". Det udregnede middeltryk gemmes i en liste kaldet middeltrykliste.

EKG-signal

Implementeringen af EKG-signalet er i dette projekt ikke sket. Dette er ikke implementeret, da der ikke har været tid til dette. Hvis der dog havde været tid til dette, ville Semesterprojekt 2: EKG diagnostik, været blevet benyttet til at få grundstrukturen til at se hvordan denne implementering skulle foregå.

Iltmætning

Desuden blev ilt saturation ikke implementeret. Dermed bliver iltmætningen ikke vist for patienten. Dette er ikke blevet implementeret på grund af flere faktorer: Der er ikke nok tid, der er ikke den fornødne viden til at få implementeret dette og der haves ikke de nødvendige materialer.

Observer mønsteret

Observer/subject mønstreret bruges til at sende data op kontinueret og på en fornuftig måde. Subject har en metode til at koble en observer til, og en anden metode til at fjerne observeren.

Der er en tredje metode i subject, kaldet notify. Notify sørger for at give besked til den tilknyttet observer om opdatering. En observer er interesseret i at kende til ændringer i det subject den er tilknyttet. Beskeden om ændring kommer fra subjectet, der er tilknyttet observeren.

I dette projekt er der valgt at bruge observer/subject mønstreret til at sende data fra datalag til logiklaget. Mønstreret er også brugt til at sende data videre fra logiklaget til præsenterationslaget. Yderligere beskrivelse findes i projektdokumentationen under kapitlet Arkitektur og design under afsnittet Software design.

PUSH/PULL

PUSH/PULL er to forskellige måde at implementere Observer mønsteret på. Her bliver data enten skubbet eller trukket op. PUSH betyder at skubbe, og betyder at data bliver skubbet op, når der sker en ændring, uden der skal gives besked om at hente data. PULL trækker data op imellem lagene. I PULL gives der besked til et lag om en ændring. Herefter skal laget, der ønsker at kende til ændringen bede om ændringen, før det sendes op.

I dette projekt er det valgt at benytte PUSH til at sende data imellem lagene. Idet det er interessant at få data op til brugergrænsefladen hele tiden når der ny data fra DAQ'en. Yderligere beskrivelse findes i projektdokumentationen under kapitlet Arkitektur og design under afsnittet Software design.

Queue

For at sikre at data bliver sendt op i den rigtige rækkefølge og på en kontrolleret måde, er der i dette projekt valgt at benytte en kø. Til dette er der benyttet en indbygget funktion i Visual Studio, Queue class. Herunder ligger der to metoder, som kan bruges til at sende objekter af tal ind i den ene ende af køen og efter at køen er blevet fyldt op, er der en funktion til at fjerne det første objekt fra den anden ende af køen. Dette sørger for at der hele tiden er et flow i køen når der sendes data ind i denne.

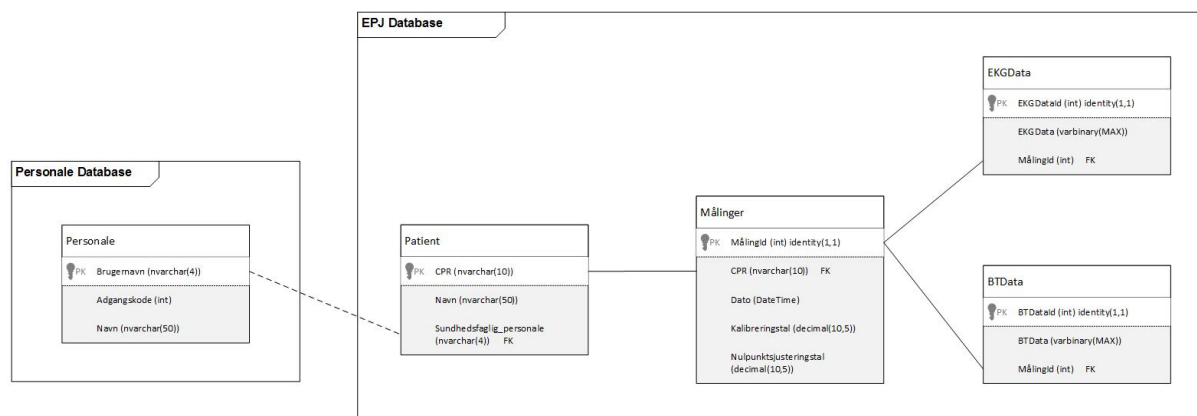
En yderligere beskrivelse af hvordan Queue klassen benyttes i koden og hvordan denne er opbygget er beskrevet i dokumentationen under Arkitektur og design under afsnittet Implementering.

Trelagsmodel

Trelagsmodellen er en software model, til at inddele sin software i tre lag, præsenterationslaget, logiklaget og datalaget. Hver lag har sit eget ansvar og funktionalitet. Præsenterationslaget kan kun tilgå logiklaget, og må kun vise data på brugergrænseflade eller indlæse data fra brugergrænsefladen. Logiklaget må tilgå både datalaget og præsenterationslaget. Logik er det midterste lag, og står for behandling af data fra præsenterationslaget og datalaget. Datalaget er det nederste lag og sørge for at sende data fra et måle apparat op til logiklaget. Datalaget står også for at gemme data i en fil eller en database.

I dette projekt er koden opdelt efter trelagsmodellen, ved at have et præsenterationslag med to brugergrænseflader, et logiklag med algoritmer til udregning af blodtryk, og et datalag der hente data fra DAQ'en og gemmer dem i en database. Yderligere beskrivelse og model over trelagsmodellens inddeling af software koden findes i projektdokumentationen under kapitlet Arkitektur og design under afsnittet Software design.

Databaser



Figur 6.10: Databaserne; Personale database og EPJ database. Her ses sammenhængen mellem tabellerne i databasen, og hvilken data der er under hver tabel.

Dette afsnit vil beskrive opbygningen af databaser og tabellerne. Da der er valgt at tage udgangspunkt i et brugsscenarie, afspejler datamodelleringen i databasen dette scenarie. Det er vigtigt at bemærke at der er en regulær opdeling af databaser og skemaer, realiseret som beskrevet nedenfor.

Personale database

Personale databasen bliver brugt til at tilgå data for personalet. Her er der tale om brugernavn og password. Disse data er lagt i databasen på forhånd, så det er altså en login-database.

Database-adresse: webhotel10.F15ST2ITS2201404669.db__ owner

EPJ database

EPJ databasen bliver brugt til at tilgå data for patienten. Her tænkes på CPR/Navn. Ligeledes gemmes der i denne database målinger, samt metadata for at kvalificere disse data. Der kunne udvides med mange flere metadata, specielt i forhold til afsnittet om fremtidigt arbejde "Datawarehouse". Det bør bemærkes der i databasen er gjort klar til EKG-data, selvom det ikke er implementeret. Patient-tabellen modsvarer vores version af en EPJ-database, hvor BTData-tabellen repræsenterer målingerne foretaget af vores system.

Database-adresse: webhotel10.F15ST2ITS2201405838.db__ owner.

Unittest

Det er blevet besluttet at lave unittest på softwaren. Unit-test forstås ved at der testes på hvert enkelt komponent i koden, i objektorienteret kode vil det sige metoderne. Man sikrer sig at metoderne returnerer det forventede. En unit test bør foregå løbende lige så frit som koden bliver udviklet, så kan man fokusere på de del-elementer som ikke virker efter hensigten.

I dette projekt blev der valgt at lave unit-test efter sidste iteration af koden, da de sidste versioner af softwaren ændrede sig meget, så alle unit-tests skulle laves forfra. Grundlaget for mange metoder ændrede sig løbende. Disse unit-tests blev lavet i koden ved at sætte breakpoints og følge metoden til ende og evaluere på hvorvidt metoden opførte sig efter hensigten. Unit-tests

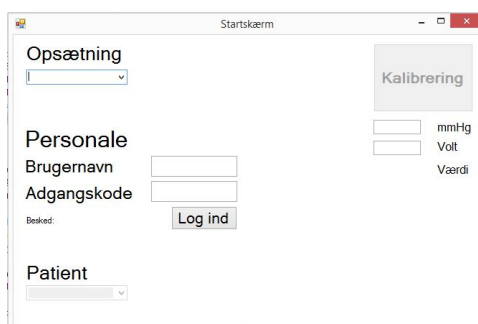
er blevet foretaget i koden, hvor der er skrevet kommentarer i koden til de metoder der er blevet testet.

6.5 Resultater og diskussion

Software

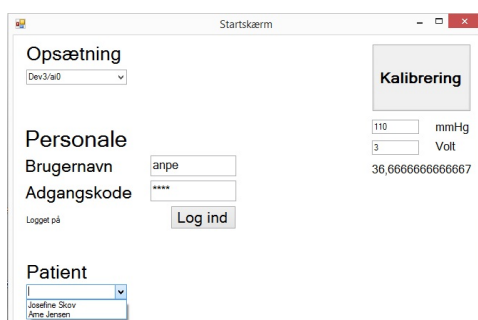
Igennem integrationstesten og accepttesten er der opnået resultater for projektet. Igennem accepttesten er der opstillet krav til hvordan blodtryksmålesystemet skal opføre sig i forhold til Use cases, og hvordan dette kommer til udtryk visuelt. Resultaterne for projektet er visuelle resultater af accepttesten, og ses i dette afsnit som screendumps.

I accepttesten er det første step for accepttest af Use case 1, er at vælge en værdi på vandsøjlen og kalibrer efter den. Herved skal den aflæste spænding og trykket i vandsøjlen indtastes. Når programmet starter, vises start skærmen neden for.



Figur 6.11: Startskærm uden indtastede login oplysninger.

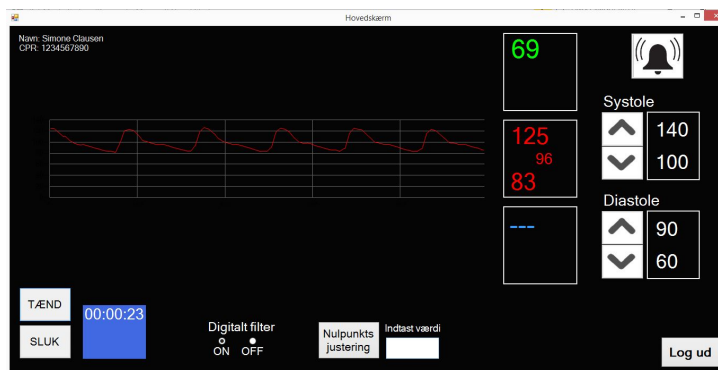
Der kan ses på figuren neden for at det er muligt at kalibrer på hovedskærmen, og at blodtryksmålesystemet selv udregner en kalibreringsværdi ud fra de indtastede oplysninger. Kalibreringsværdien udregnes først, når der er trykket på kalibreringsknappen.



Figur 6.12: Startskærm med login oplysninger og kalibreringsværdi.

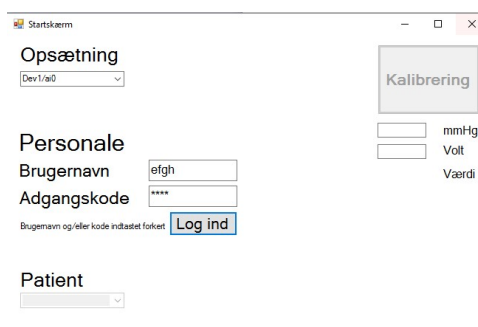
Næste step i accepttesten er for Use case 2, er at indtaste brugernavnet "anpe" og koden "1234". Derefter trykkes der på login og vælge patient i patient dropdown. Inden dette er sket er der valgt den port, som DAQ'en er tilkoblet computeren i, og der er logget ind på VPN. På figuren oven for, kan der ses at der er indskrevet brugernavn og adgangskode for anpe, og der vises hvilke patienter anpe kan tilgå i Patient-dropdown. Efter at have valgt en patient

i patient dropdown, kommer hovedskærmen frem. Her skal der trykkes på tænd-knappen, og derefter bliver grafen for blodtrykket vist, samt værdierne for systolen, diastolen, middeltrykket og pulsen. Det ses på figuren neden for at blodtrykket bliver vidst med filteret på, og der vises en puls værdi på 69, samt en systole på 125, en diastole på 83 og et middeltryk på 96.



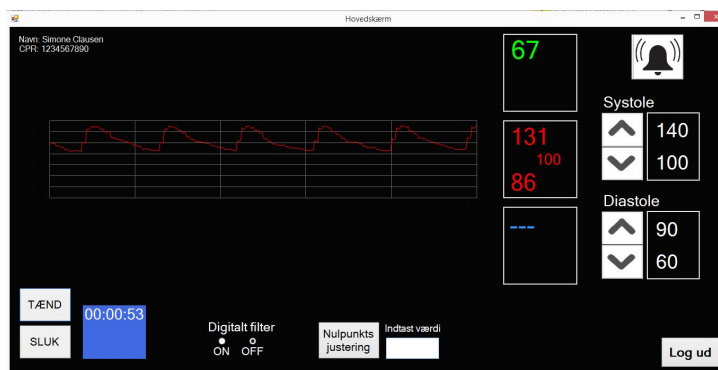
Figur 6.13: Blodtryksmåling med det digitale slået til.

Blodtryksmålesystemet skal også kunne angive hvis brugernavnet/adgangskoden ikke findes i databasen. Herved er der i accepttesten for Use case 2, indtastet brugernavnet efgh og koden 1234. Ingen af disse findes i personale databasen, og kan herved ikke bruges til at logge ind med. Som det ses på figuren neden for, er det heller ikke muligt at logge ind, og programmet giver besked om, at brugernavn og/ eller kode er indtastet forkert.



Figur 6.14: Brugerneavn/kode er indtastet forkert.

I blodtryksmålesystemet skal det være muligt at kunne slå filteret til og fra, dette bliver testet i accepttesten ved at klikke i radiobutton for digitalt filter off. På figuren neden for, ses at det blodtrykssignalet er forvrænget, da der er kommet mere støj på signalet, herved kan det konkluderes at filteret virker.



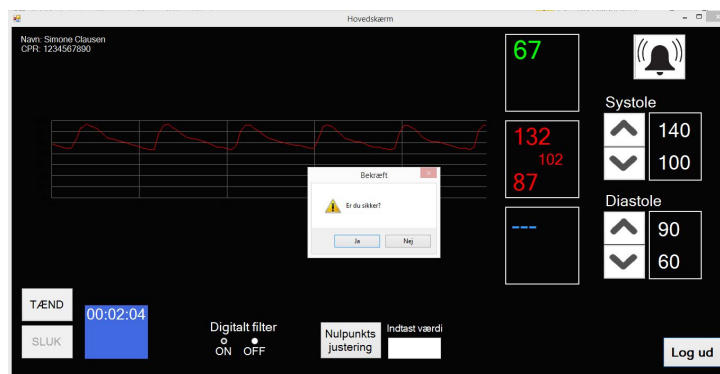
Figur 6.15: Blodtrykssiganelet uden digitalt filter.

I accepttesten skal blodtryksmålesystemet kunne melde alarm når grænseværdierne overskrides, og grænseværdierne skal kunne ændres på hovedskærmen. På figuren neden for, ses det at grænseværdierne for systolen op og ned, er ændret til 129/89, hvor de før var sat til normal værdierne for systolen, 140/100. Blodtrykssignalet har en systoleværdi på 132, hvilket er over den nye grænseværdi på 129, og derved går alarmen i gang, som det ses med den røde alarm klokke øverst i højre hjørne af hovedskærmen.



Figur 6.16: Grænseværdi for systole ændret og alarm igangsat.

I accepttesten for Use case 4, skal der trykkes på sluk, derefter log ud og til sidst ja på pop-up vinduet. På figuren neden for, ses der at der er trykket på sluk-knappen, da den er blevet gennemsigtig, og kun tænd-knappen der kan trykkes. Log ud er blevet trykket for at bekræft pop-up vinduet er kommet frem. Herved virker log ud knappen som den skal, og der bliver gemt målingen for patienten i EPJ-databasen. Blodtryksmålingen for patienten bliver gemt løbende i EPJ-databasen.



Figur 6.17: Log ud af blodtryksmålesystemet.

Fra de foregående figur kan det konkluderes at blodtryksmålesystemet lever op til accepttestens krav omkring, at kunne vise et blodtrykssignal kontinuert, samt slå et digitalt filter til og fra.

Blodtryksmålesystemet lever også op til kravet omkring at kunne kalibrere systemet. Dette gøres på startskærmen. Kravet til at systemet kan nulpunktsjusteres er ligeledes opfyldt, denne sker på hovedskærmen. Det kan diskuteres om det var smartere at systemet selv skal kunne indlæse værdien for nulpunktjustering, og derefter bare lægge denne værdi til blodtrykssignalet, når der trykkes på nulpunktsjusterings knappen, end at man selv skal aflæse værdien. Dog lever nulpunktsjusterings knappen, op til kravet omkring at nulpunktsjustere blodtryksmålesystemet, da alle blodtrykssignalsværdierne bliver nulpunktsjusteret efter den indtastet nulpunktsværdi.

Grænseværdierne for blodtryksmålesystemet kan også ændres, herved er dette krav opfyldt, og der kommer en alarm når værdierne for enten systolen eller diastolen overskrider de satte grænseværdier.

Alarmen kan også udsættes i et minut, hvilket gøres ved at trykke på alarmknappen oppe i højre hjørne af hovedskærmen.

En anden ting som blodtryksmålesystemet også kan er at vise en timer, som starter når man har trykket på start knappen, og stopper når der trykkes på sluk knappen. Det kan diskuteres hvor smart det er, om timeren skal nulstille, når der trykkes på start knappen igen. Der er valgt i dette projekt at timeren starter fra det stoppet tidspunkt, når tænd bliver trykket igen, dette er valgt fordi, at det giver bedst mening at man skal kunne starte fra hvor man slap. Dog giver den funktion med at starte timeren igen, ikke så meget mening ude i den virkelige verden, da anæstesi sygeplejerskerne ikke starter målingen igen, efter at have stoppet målingen.

Det kan diskuteres om brugeren skal have lov til at kalibrer på startskærmen, da kalibrering skal fortages af en servicemedarbejder. Herved skulle der kun have været log ind på startskærmen, og være lavet et servicevindue, som kun kan betjenes af servicemedarbejderen. Der er taget højde for denne problemstilling i koden, ved at have lavet en config fil, hvor kalibreringstallet kan ændres. Det betyder, at hvis det sundhedsfagligt personale springer over at kalibrere på startskærmen kalibrerer systemet automatisk efter kalibreringstallet i config filen. Meningen med config filen, er at det kun er servicepersonalet der skal kunne tilgå denne fil, og kunne ændre kalibreringstallet.

Hardware

De vigtigste resultater på hardware delen inkluderer signal forstærkning og signal filtrering. Til simulering af transduceren blev der generet et signal på 20mV fra Analog Discovery og det forventet resultat var at filtret dæmper frekvenser over 50Hz. De målte målinger fra hhv. forstærkeren og filtret ser således ud:

Frekvens [Hz]	Gain. [V] Peak to peak
1	16,308
10	17,512
25	17,558
49	17,512
70	17,258
100	17,308
250	16,278
500	15.051

Som det ses på tabellen har operationsforstærkeren levet op til det forudsete udfald. De 20mV bliver forstærket. Den gainKilde, som er beregnet i teorien var 800 og det vi har målt er i fint tråd med det teoretisk gain.

$$output_{INA114} = 20 \cdot 10^{-3} \cdot 800 = 16 \quad (6.12)$$

Den anden måling som skulle foretages var at se om filtret lever op til det krav der er opstillet, nemlig en båndbredde på 50Hz der minimum giver 20 dB dæmpning på en dekade. Resultatet af denne måling er som følgende:

Frekvens [Hz]	Peak to peak [V]
1	13,504
10	13,220
25	12,412
49	10,644
70	6,658
100	3,662
250	0,666
500	0,220

Om filtret har levet op til det forventet resultat kan tjekkes som følgende:

$$G_{dB} = 20 \cdot \log \left(\frac{V}{V_{out}} \right) = 20 \cdot \left(\frac{16,308}{0,220V} \right) = -37,234dB \quad (6.13)$$

Kravet til filtret var at ved frekvens 500Hz skal signalet dæmpes med 40dB på en dekade. Overstående beregning giver en dæmpning på -37,234 dB, hvilket ikke er helt ved siden af. Afvigelsen kan skyldes anvendelsen af ikke-ideelle komponenter.

Ved knækfrekvensen skal dæmpning være 3dB og det kan regnes som følgende:

$$G_{dB} = 20 \cdot \log \left(\frac{V}{V_{out}} \right) = 20 \cdot \log \left(\frac{10,664V}{15,051V} \right) = -3,009dB \quad (6.14)$$

Ud fra de overstående beregninger, kan det ses at både operationsforstærkeren og filtret levet op til kravene tilnærmelsesvis.

6.6 Udviklingsværktøjer

Til udviklingen af produktet, og dermed igennem hele projektarbejdet, er der blevet brugt en række udviklingsværktøjer. I dette afsnit er disse udviklingsværktøjer beskrevet yderligere.

Microsoft Visio 2010

Tegneværktøjet Microsoft Visio, er blevet anvendt i forbindelse med design af både UML og SysML diagrammer. Microsoft Visio er det oplagte valg til at designe disse diagrammer, da programmet søger for at diagrammerne får et enkelt udseende og tydeligt kommunikerer til læseren, hvad diagrammerne vil vise.

Visual Studio 2013

Visual Studio 2013 er et udviklingsværktøj, designet til Microsoft .Net-frameworket. Dette udviklingsværktøj er specielt velegnet til design af brugergrænseflader, som i dette projekt er specifikt brugbart, da der skal udarbejdes et Windows Forms program.

NI Multisim 13.0 med Ultiboard 13.0

Til at designe hardwaren er Multisim med Ultiboard blevet benyttet. Multisim er et program der kan bruges ved kredsløbsdesign, da komponenterne her er lette at sætte i forhold til hinanden, da programmet indeholder samtlige komponenter. Ultiboard bruges til at designe printet, der skal bruges til at realisere hardwaren, der er blevet designet i Multisim. Ved at benytte Ultiboard undgås det at benytte et fumlebræt eller veroboard, hvor komponenterne skal kobles sammen med ledning. Ved et print der designs med Ultiboard vil forbindelserne mellem komponenterne være direkte implementeret på printet og komponenterne vil derfor være nemme at implementere på printet.

NI-DAQmx

NI-DAQmx, også omtalt NI-DAQ/DAQ, er et værktøj som er blevet udarbejdet af National Instrument. Dette værktøj anvendes til at indsamle det indkommende signal. Dette signal kommer fra hardwaren. NI-DAQ omdanner desuden signalet fra et analogt signal til at være et digitalt signal der kan bruges i koden.

Analog Discovery fra Digilent and Analog Devices

Analog Discovery er i projektet den der benyttes om en waveform-generator. Det er derfor Analog Discovery der benyttes til at teste systemet. Dette sker ved at der sendes et blodtrykssignal fra Physionet.org ind i systemet, som systemet derefter skal behandle. Desuden er oscilloskop-funktionen i waveforms blevet benyttet under test af hardware.

6.7 Opnåede erfaringer

På baggrund af vores sundhedsfaglige viden og de opsatte rammer for projektet, er der blevet opstillet en projektformulering, som er løst ved hjælp af viden omkring programmering, analog og digital signalbehandling, SysML og UML. Arbejdet har styrket vores tværfaglige kompetencer og vi har formået at skabe en teknisk løsning på et sundhedsfagligt problem.

Kompetencerne inden for digital signalbehandling, er indenfor dette projekt opnået ved at udarbejde et moving average lavpasfilter. Inden det blev klart at dette lavpasfilter skulle benyttes, er flere forskellige filtre blevet afprøvet. Disse filtre er ved brug af programmet MatLab blevet afprøvet, et butterworth IIR (Infinite impulse filter) lavpas filter her blev udarbejdet. Det var dog et FIR (Finite impulse filter) lavpasfilter der skulle benyttes, hvilket førte frem til at moving average filteret blev afprøvet, hvilket virkede efter hensigt.

Programmerings koden til filteret skulle herefter udarbejdes, her blev der undersøgt forskellige metoder der kunne benyttes via søgninger på internettet, hvilket sammen med at "snakke og tegne" koden førte frem til at koden kunne skrives til det digitale filter. Realiseringen af lavpasfilteret forbedrede derfor ligeledes programmeringskompetencerne. Det at "snakke og tegne" koden, blev brugt ved alle problemstillingerne i programmeringen, for at finde frem til den tilgang der skulle bruges for at løse problemet. Bl.a. ved udarbejdelse af metoden til hvordan der skulle holdes styr på den data der skulle sendes op, blev koden "snakket og tegnet" igennem inden. Idet blodtrykket skulle vises løbende, skulle der være en kømetode, som kunne indhente og udsende data løbende og samtidig holde styr på data i køen.

Andre software kompetence er ligeledes blevet opnået, ved at benytte forskellige mønstre og kode opbygninger, hvor netop kombinationen af disse faktorer har ført til en bredere forståelse i programmering og hvordan et projekt logisk kan bygges godt op. Herunder hvor de forskellige metoder må ligge og hvordan klasserne må snakke sammen.

Data der skulle sendes ind i kodes, skulle først igennem den udarbejdede hardware. Under udarbejdelsen af hardware produktet er kompetencerne inden for analog signalbehandling, blevet forbedret. Ud fra problemstillingen og projektformuleringen vidste vi hvordan signalet skulle sendes ind, her skulle kredsløbet, der skulle praktisere dette, først designes. En forstærker og et lavpasfilter skulle designes, hvilket blev gjort i forskellige fase, først igennem forskellige beregninger, som kunne føre til at bestemme komponenterne der skulle benyttes ved forstærkeren og lavpasfilteret. Herefter kunne kredsløbet designes i multisim og på fumlebræt for at se om det analyserede kredsløbet kunne realiseres.

Det var lærerigt at give reviews til andre grupper, da vi her fik muligheden for at reflektere over, hvordan vi selv havde arbejdet med projektet og samtidig fik vi et indblik i, hvordan de andre grupper havde valgt at gribe opgaven an. Der blev udført to reviews, et review på kravspecifikation og accepttest og et review på design og specifikation af software og hardware. Desværre har der gennem projektførløbet været en del forvirring på tværs af grupperne omkring diverse krav og fremgangsmåder vedrørende projektførelsen, hvilket gjorde review møderne mindre udbytterige end de kunne have været.

Projektstyringen og ledelsen må vi indse har været for svag i dette projekt. Rigtigt ville vi have benyttet SCRUM, men dette kom ikke rigtig op og køre, så opgaverne har været mere løst opsatte. Hvor vi igennem SCRUM kunne have haft identificeret mere klare opgaver og sætte en fast deadline for, hvornår opgaven skulle være færdig. Ligeledes kunne opgaverne her prioriteres sådan det også blev identificeret hvor stor opgaven er og hvor høj prioritering denne opgave har. Det må dog også erkendes at fordelingen af ressourcerne ikke har været optimalt.

6.8 Fremtidigt arbejde

I dette projekt har der været flere overvejelser om hvorledes vi ønskede vores slutprodukt skulle se ud og hvilke funktioner vi efterstræbte at opfylde. Vores besøg på Herning sygehus gav os

et indblik i hvordan en invasiv blodtryksmåler fungerer i praksis og hvilke funktioner der er indbygget heri. Vi ønskede altså at basere vores system ud fra et virkeligt brugsscenarie. Dette brugte vi som udgangspunkt for vores system og vi udtænkte derfor en ide og opstillede et design, som vi ville udarbejde i vores projekt. Vores produkt skulle slutteligt kunne vise både EKG, systolisk samt diastolisk blodtryk og iltmætning. Alle værdier skulle udskrives i en graf og talværdierne vises på brugergrænsefladen.

EKG-signal

EKG-signalet skulle simuleres ved at sende et EKG-signal fra physionet.org via analog discovery ind i en anden port i DAQ'en, dette signal skulle så på lige fod med blodtryks-data præsenteres visuelt. Ligeledes skulle der være mulighed for at disse data kunne gemmes i databasen.

Iltmætning

Iltmætning ønskede vi at implementere i vores system, så dette, ligeledes som blodtrykket og EKG, ville blive præsenteret visuelt. For at få vist iltmætning ville vi benytte et pulsoxymeter. Pulsoxymeteret sættes på fingeren og virker ved at gennemlyse fingeren med infrarødt lys. Herved måles iltmætningen i blodet. Disse data kunne via USB overføres til programmet.

EPJ

Fremtidigt ville vi implementere vores system op mod eksisterende EPJ systemer, da vores system allerede er baseret på et virkeligt brugsscenarie, forventer vi at en implementation ville være relativt enkel. Vores patientdata ville i stedet for vores database være hentet fra EPJ. Ligeledes vil vores logindata for plejepersonalet være hentet enten i Active Directory (netværket) eller i medarbejderdatabasen. Blodtryksdata vil stadig skulle gemmes separat, men som vi kommer ind på i næste afsnit omkring datawarehouse, så skal data være kvalificeret af en række metadata.

Datawarehouse

Det blev fra projektets start talt om at lave et datawarehouse, realiseret med et slags mock-up datawarehouse til analysering af blodtryksmålinger. Et datawarehouse er en samling af data fra forskellige kilder, der er sorteret efter emne, i vores tilfælde ville det være eksempelvis være blodtryksmåledata. Dette emne ville så være kvalificeret af en række dimensioner, som vil være repræsenteret af metadata i vores database.

Metadata / dimensioner kunne være ting som:

- Aldersgruppe eller alder
- Køn
- Datotid
- SKS-koder (Sundhedsvæsnets Klassifikations System)
- Behandlingssteder/Afdelinger
- Andre metadata fra EPJ

Et datawarehouse fungerer absolut bedst når der er akkumuleret mange data, da det giver et mere retvisende billede af tendenser.

Kapitel 7

Konklusion

Vi kan i dette projekt konkludere, at der er blevet udarbejdet et system, som kan tilsluttes et væskefyldt kateter og vise en blodtrykskurve på en computerskærm. Igennem dette projekt ønskede vi, at opstille et system, som afspejler et hverdagsscenarie og efterlever de funktionaliteter, som vi fik fortalt og beskrevet af en anæstesisygeplejerske fra Herning Sygehus. Overordnet set, ønskede vi altså at opstille to systemer tilsvarende dem, som benyttes på sygehuse i dag. Et som skulle simulere EPJ systemet og et som skulle simulere den monitor, som bruges til afbildning af blodtryk, EKG og iltmætning.

I vores system har vi implementeret de to systemer, som to skærme. Den første skærm, som vises i vores program er startskærmen, det er her personalet logger ind og vælger patienten. Derudover er det også her vi kalibrerer. Denne funktion skal udføres af en servicemedarbejder og ville i et hverdagsscenarie ikke ske i EPJ systemet. Når der er logget ind kommer vi til hovedskærmen. Her vises patientens navn og CPR og det sundhedsfaglige personale kan starte blodtryksmålingen. I vores system var det tidsmæssigt og ressourcemæssig ikke mulig, at få implementeret EKG og iltmætning og vi viser derfor kun blodtrykket, med systolisk, diastolisk og middeltryks værdier. På denne skærm har vi valgt, at nulpunktjustering skal kunne foretages, men er bevidste om, at dette ville ske automatisk og altså ikke er vist på en blodtryksmåler i praksis. Vi har endvidere udarbejdet to databaser, en EPJ database og en personale database, som afspejler det EPJ system, der bruges i sundhedssektoren i dag.

Vi kan efter denne projekt proces konkludere, at vi har opnået det overordnede resultat og mål, som vi havde for vores system. Vi har igennem interaktion mellem hardware og software fået vist et blodtrykssignal og der udskives systolisk og diastolisk tryk, som er indenfor de grænseværdier, der ses ved et normalt blodtryk.

I forbindelse med proces ledelse og deltagelse, har vi i gruppen lært vigtigheden i at have en leder, som kan holde et overordnet overblik og tage styring og lederskab når dette er nødvendigt. Vi har i gruppen været seks mand, og det er derfor vigtig, at en fra holdet kan samle trådende og sørge for et fælleskab i gruppen, hvor alle yder samme indsats for samarbejdet og projektet. Vi har derfor også været nødsaget til samle gruppen og udnævne en ny leder, som turde tage de valg og det ansvar, som følger med lederrollen. Samlet set har vi i dette projekt opnået ny viden og draget nyttes værdige erfaringer, som vi kan tage med i et videre projekt arbejde.

Litteratur

7.1 Referencer

Billeder

- [1] Kredsløb og de vigtigste organer, fc.vucnordjylland.dk, læst d. 8. december 2015, <http://fc.vucnordjylland.dk/Flexundervisning/FOV1-000D35E9/FOV1-000948AA/Modul%204/Inspiration/I062B76D7>

Bøger

Internetsider

- [2] hjerte (mennesket), denstoredanske.dk, læst. 12. december 2015, [http://www.denstoredanske.dk/Krop,_psyke_og_sundhed/Sundhedsvidenskab/Sammenlignende_anatomi_og_fysiologi/hjerte/hjerte_\(Mennesket\)](http://www.denstoredanske.dk/Krop,_psyke_og_sundhed/Sundhedsvidenskab/Sammenlignende_anatomi_og_fysiologi/hjerte/hjerte_(Mennesket))
- [3] Blodtryk, Wikipedia.org, læst d. 7. december 2015, <https://da.wikipedia.org/wiki/Blodtryk>
- [4] Hypertension, Sundhed.dk, skrevet 4. august 2015, læst d. 7. december 2015, <https://www.sundhed.dk/sundhedsfaglig/laegehaandbogen/hjerte-kar/tilstande-og-sygdomme/oevrige-sygdomme/hypertension/>
- [5] Forskelle mellem systemiske & pulmonale cirkulation, Michael Kristensen, skrevet d. 24. maj 2015, læst d. 8. december 2015, <http://www.sundhedguide.com/2/2015/05/forskelle-mellem-systemiske-pulmonale-cirkulation.html>
- [6] Lavt blodtryk, Thomas Baastrup Piper, læst d. 13. december 2015, [http://www.sundhedsguiden.dk/da/temaer/alle-temaer/blodtryk/blodtryk/lavt-blodtryk-\(hypotoni-hypotension\)/](http://www.sundhedsguiden.dk/da/temaer/alle-temaer/blodtryk/blodtryk/lavt-blodtryk-(hypotoni-hypotension)/)
- [7] UML, Wikipedia.org, Tim Ottinger og Jeff Langr, læst d. 9. december 2015, <http://da.wikipedia.org/wiki/UML>
- [8] FURPS+, læst d. 9. december 2015, <http://agileinaflash.blogspot.dk/2009/04/furps.html>
- [9] Pivotal Tracker, <https://www.pivotaltracker.com/>
- [10] Sallen–Key topology, Wikipedia.org, læst d. 15. december 2015, https://en.wikipedia.org/wiki/Sallen%E2%80%93Key_topology#Application:_Low-pass_filter

Dokumenter

- [11] A practical guide to SysML, Sandford Friedenthal, Alan Moore og Rich Steiner, s. 29-60.