



AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI
2. SEMESTERPROJEKT

Dokumentation

Gruppe 1

Lise Skytte Brodersen (201407432)
Mads Fryland Jørgensen (201403827)
Albert Jakob Fredshavn (201480425)
Malene Cecilie Mikkelsen (201405722)
Mohamed Hussein Mohamed (201370525)
Sara-Sofie Staub Kirkeby (201406211)
Martin Banasik (201408398)
Cecilie Ammizbøll Aarøe (201208778)

Vejleder

Studentervejleder
Lars Mortensen
Aarhus Universitet

13. maj 2015

Gruppemedlemmer

Lise Skytte Brodersen (201407432)	Dato
Mads Fryland Jørgensen (201403827)	Dato
Albert Jakob Fredshavn (201480425)	Dato
Malene Cecilie Mikkelsen (201405722)	Dato
Mohamed Hussein Mohamed (201370525)	Dato
Sara-sofie Staub Kirkeby (201406211)	Dato
Martin Banasik (201408398)	Dato
Cecilie Ammitzbøll Aarøe (201208778)	Dato

Vejleder

Lars Mortensen	Dato
----------------	------

Ordliste

Ord	Forklaring
(F)URPS+	Et akronym, der repræsenterer en model til klassificering af softwarens kvalitet
GUI	Graphical User Interface (Grafisk brugergrænseflade)
EKG	Elektrokardiografi
UC	Use Case
AT	Accepttest
KS	Kravspecifikation
DAQ	Data acquisition
Fully dressed Use Case	Fully Dressed Use Cases er Use cases med flere detaljer, mere dybdegående og struktureret
BDD	Blok definition diagram
IBD	Internal blok diagram
UML	Unified modeling language

Indholdsfortegnelse

Ordliste	iii
Kapitel 1 Indledning	1
Kapitel 2 Kravspecifikation	3
2.1 Indledning	3
2.2 Funktionelle krav	3
2.2.1 Aktør-kontekstdiagram	3
2.2.2 Aktørbeskrivelse	4
2.2.3 Use case-diagram	4
2.2.4 Use Cases	4
2.3 Ikke-funktionelle krav	8
2.3.1 (F)URPS+	8
Kapitel 3 Design	11
3.1 Indledning	11
3.2 Hardware arkitektur	11
3.2.1 Grænseflader	12
3.3 Software arkitektur	13
3.3.1 GUI	14
3.3.2 UML klassesdiagram	17
3.3.3 Applikationsmodel	18
Kapitel 4 Accepttest	25
4.1 Accepttest af Use Cases	25
4.1.1 Use Case 1	25
4.1.2 Use Case 2	26
4.1.3 Use Case 3	26
4.1.4 Use Case 4	27
4.1.5 Use Case 5	27
4.2 Accepttest af ikke-funktionelle krav	28
Bilag	31

Indledning 1

Denne dokumentation udgør baggrunden for projekt rapporten omkring EKG-måling af atrieflimren. I dokumentationen beskrives først og fremmest hovedscenariet, gennem udarbejdelse af kravspecifikation.

Kravspecifikationen består en beskrivelse af projektets funktionelle krav. Beskrivelse af aktører, samt hvordan de interagerer, derudover også beskrivelse af de relevante use-cases, samt systemets undtagelser. Desuden er systemets ikke-funktionelle krav også beskrevet, dette gennem systembeskrivelsesmetoden "(F)URPS+".

Ydermere formidles en detaljeret beskrivelse af det komplette systems design, bestående af hardware og software arkitektur. Hardwaren arkitekturen beskriver den grundlæggende opstilling af systemet, samt tilhørende grænseflader. Softwaren er beskrevet via en gennemgang af hovedforløbet illustreret ved skitser af GUI'en, samt diverse relevante systembeskrivelsesmodeller.

Afslutningsvis foretages en fyldestgørende accepttest, som primært har til formål at teste de opstillede use-cases, samt systemets ikke-funktionelle krav. Accepttesten beskrives efterfølgende kronologisk fra use case 1 til ikke-funktionelle krav, for at i sidste ende at kunne dokumentere EKG-systemets funktionalitet.

Kravspekifikation 2

Version	Dato	Ansvarlig	Beskrivelse
1.0	18-03-2015	LSB, AJF og MFJ	Påbegyndt tilrettelse af UC1 og UC2, aktør-kontekst samt funktionellekrav i forhold til den valgte sygdom, Atrieflimren
1.1	26-03-2015	LSB, AJF, MFJ og CAA	KS er færdigskrevet og klar til review
2.0	09-04-2015	LSB, AJF, MFJ og CAA	Rettet i forhold til review-kommentarer
Tekst	Tekst	Tekst	Tekst.

2.1 Indledning

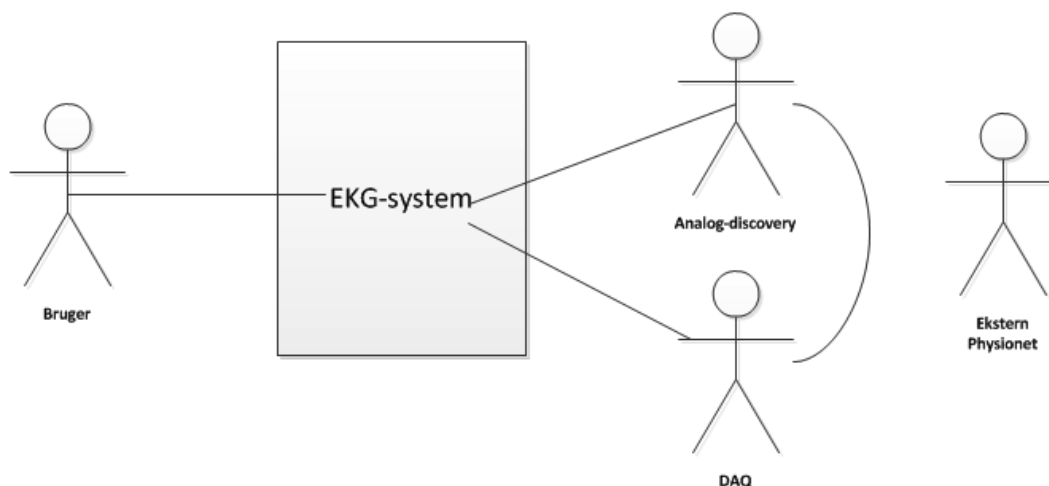
Kravspekifikationen vil beskrive, ud fra en række modeller, hvordan EKG-systemet fungerer. Helt generelt er EKG-måling en simpel metode, til at måle hjertets elektriske aktivitet via elektroder, som registrerer elektriske impulser, placeret på huden. Ud fra disse impulser dannes en graf, som benyttes til at analysere hjertets funktionalitet ud fra P-, Q-, R-, S- og T-takkerne, og dermed konkludere om den pågældende patient har et raskt eller sygt hjerte, samt hvilken sygdom der er tale om. Helt specifikt for denne opgave er formålet, at identificere sygdommen atrieflimmer via et virtuelt EKG-signal.

2.2 Funktionelle krav

De funktionelle krav vil nedenstående beskrives ud fra Aktør-kontekstdiagram, aktørbeskrivelse, Use Cases samt Use Case diagram.

2.2.1 Aktør-kontekstdiagram

Data hentes ned fra den ekstern aktør, physionet, og via Analog-discovery omdannes csv-filens data til et analogt signal, der sendes til EKG-systemet. Ud fra disse data danner EKG-systemet en graf. Programmet detekterer markørudsving i EKG-grafen, som derefter valideres og analyseres af brugeren.



Figur 2.1: Aktør-kontekstdiagram

2.2.2 Aktørbeskrivelse

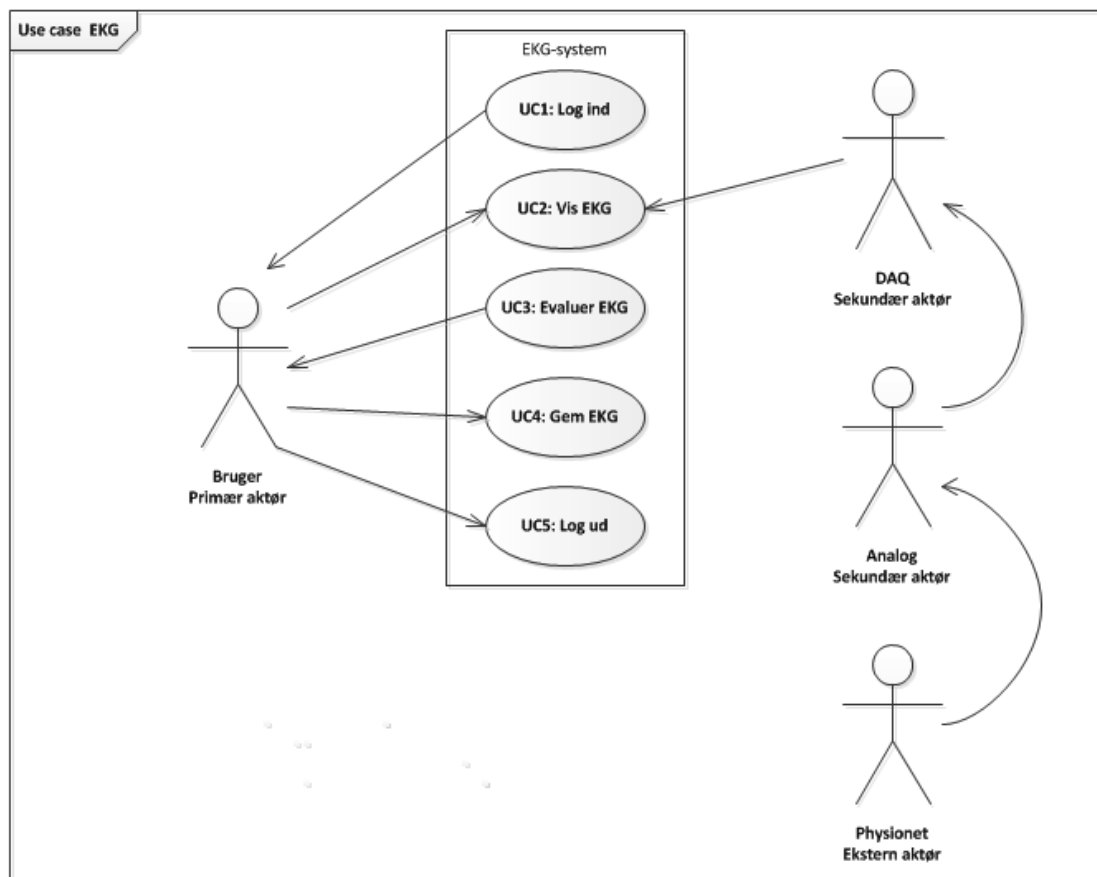
Aktørnavn	Type	Beskrivelse
Bruger	Primær	Brugeren er den aktør, der ønsker at foretage målingerne, som omfatter EKG samt diagnosticering af atrieflimmer. Brugeren er en person, der har kendskab til EKG-systemet. Fx sundhedsfaglig personale
Analog-discovery	Sekundær	Analog-discovery omdanner data fra den eksterne aktør, physionet, til et analog signal
DAQ	Sekundær	DAQ'en omdanner det analoge signal fra analog-discovery til et digitalt signal, som EKG-systemet kan generere en graf ud fra
Physionet	Ekstern	Physionet er en database, hvor der ligger mange forskellige EKG-signaler. Det er ud fra disse EKG-signaler, virtuelle patienter skabes.

Tabel 2.2: Aktørbeskrivelse

2.2.3 Use case-diagram

Brugeren, den primære aktør bliver bedt om sit log ind, inden EKG-vinduet vises. Brugeren vælger indstillinger og trykker på "start-knappen. EKG-dataerne fra den eksterne aktør, Physionet, behandles i Analog samt i DAQ'en, de sekundære aktør, hvor efter data vises som en EKG-graf i EKG-vinduet. Brugeren kan ud fra denne graf evaluere EKG-signalet i forhold til at diagnosticere atrieflimmer. Brugeren gemmer EKG-målingen i databasen og logger ud.

2.2.4 Use Cases



Figur 2.2: Use case-diagram

Use Case 1

Navn	Log ind
Use case ID	1
Samtidige forløb	1
Primær aktør	Brugeren
Initialisere	Brugeren ønsker at logge ind
Forudsætninger	At der er logget ud efter en tidligere måling
Resultat	Brugeren bliver logget på og kan foretage en måling
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren indtaster username samt password 2. Brugeren trykker på "Login-knappen". Login-vinduet lukkes ned mens CPR-vinduet åbnes [2.a Username eller password er forkert]

Undtagelser	2a. Besked vises på skærmen med tekst, der informerer om, at username eller password er forkert. Der fortsættes i UC1 ved punkt 1
-------------	---

Tabel 2.3: Fully dressed Use Case 1.

Use Case 2

Navn	Vis EKG
Use case ID	2
Samtidige forløb	1
Primær aktør	Brugeren
Sekundær aktør	Analog
Sekundær aktør	DAQ
Ekstern aktør	Physionet
Initialisere	Brugeren ønsker at foretage en EKG-måling
Forudsætninger	Brugeren er logget ind og EKG-vinduet er vist samt Analog og DAQ'en er koblet til og data er hentet ned
Resultat	EKG-graf bliver vist
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren indtaster virtuel patients CPR-nummer [1.a <i>CPR-nummeret findes ikke</i>] 2. Brugeren vælger indstillinger [2.a <i>Brugeren er tilfreds med default-indstillingerne</i>] 3. Målingen startes ved at trykke på "Start" 4. EKG-data illustreres på en graf
Undtagelser	<ol style="list-style-type: none"> 1a. CPR-nummeret findes ikke. Besked vises på skærmen med tekst, der informerer om, at CPR- nummeret ikke er gyldigt. UC2 startes forfra med nyt CPR-nummer 2a. Der blev ikke ændret i default-indstillingerne. Der fortsættes ved punkt 2 i hovedforløbet med default indstillingerne

Tabel 2.4: Fully dressed Use Case 2.

Use Case 3

Navn	Evaluer EKG
Use case ID	3
Samtidige forløb	1
Primær aktør	Brugeren
Initialisere	Use Case 2 er gennemført
Resultat	Brugeren kan ud fra EKG-graf diagnosticere sygdommen atrieflimmer
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren validere programmets analyse af EKG-signalet 2. Brugeren stiller diagnosen atrieflimmer [2.a <i>Atriefrekvensen er ikke i intervallet 220-300 pr. minut</i>]
Undtagelser	2a. Det er ikke muligt at diagnosticere atrieflimmer ud fra grafen. Use case 3 afsluttes og Use case 2 gentages med evt. nye tidsindstillinger

Tabel 2.5: Fully dressed Use Case 3.

Use Case 4

Navn	Gem EKG
Use case ID	4
Samtidige forløb	1
Primær aktør	Brugeren
Initialisere	Brugeren ønsker at gemme EKG i databasen
Forudsætninger	Use case 3 er gennemført
Resultat	EKG er gemt i databasen
Hovedforløb	<ol style="list-style-type: none"> 1. Brugeren trykker på "Gem-knappen". En messagebox kommer frem med besked om at data er gemt 2. Brugeren trykker på "Ok"-knappen for at lukke messageboxen og EKG-vinduet vises igen
Undtagelser	

Tabel 2.6: Fully dressed Use Case 4.

Use Case 5

Navn	Log ud
Use case ID	5
Samtidige forløb	1
Primær aktør	Brugeren
Initialisere	Brugeren ønsker at logge ud
Forudsætninger	Der skal være logget ind
Resultat	Brugeren bliver logget ud, og EKG-vinduet lukkes og login-vinduet fremkommer
Hovedforløb	1. Brugeren trykker på "log ud-knappen" og EKG-vinduet lukkes, mens login-vinduet fremkommer
Undtagelser	

*Tabel 2.7: Fully dressed Use Case 5.***2.3 Ikke-funktionelle krav**

De ikke-funktionelle krav er udarbejdet ved brug af (F)URPS+. De er alle prioriteret ved MoSCoW metoden - Must (skal være med), Should (bør være med, hvis muligt), Could (kunne have med, hvis det ikke influerer på andet), Won't/Would (ikke med nu, men med i fremtidige opdateringer).

2.3.1 (F)URPS+

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

Usability

- (M) Brugeren skal kunne starte en default-måling maksimalt 20 sek. efter opstart af programmet
- (M) Brugeren skal have mulighed for at ændre tidsintervallet før målingerne foretages
- (M) Login-vinduet skal indholde en "login-knap til at logge på og få vist EKG-vinduet
- (M) EKG-vinduet skal indeholde en "start-knap til at igangsætte målingerne
- (M) EKG-vinduet skal indeholde en "stop-knap til at afslutte målingerne før den valgte tid
- (M) EKG-vinduet skal indeholde en "log ud-knap

- (M) EKG-vinduet skal indeholde en "gem-knap
- (M) Information-vinduet skal indeholde en "gem-knap
- (M) Målingen stopper automatisk efter det valgte tidsinterval

Reliability

- (M) Systemet skal have en effektiv MTBF (Mean Time Between Failure) på 20 minutter og en MTTR (Mean Time To Restore) på 1 minut.

$$Availability = \frac{MTBF}{MTBF + MTTR} = \frac{20}{20 + 1} = 0,952 = 95,2\% \quad (2.1)$$

Performance

- (M) Der skal vises en EKG-graf i EKG-vinduet, hvor spænding vises op af y-aksen (-1V til 1V) og tiden på x-aksen
- (M) Grafen skal være scrollbar på x-aksen, så brugeren selv ved brug af musen kan vælge det udsnit af grafen, der skal vises mere detaljeret
- (M) Skal tage en sample over et brugerbestemt interval, hvor frekvensen er tilpasset målingerne, således at grafen er analyserbar

Supportability

- (M) Softwaren er opbygget af trelagsmodellen

Version	Dato	Ansvarlig	Beskrivelse
1.0	15-04-2015	SSK	Udkast til BDD, IBD
1.1	23-04-2015	LSB og AJF	Udkast for domænemodel og sekvensdiagrammer for hver UC
1.2	27-04-2015	LSB og MFJ	Udkast til klassediagrammer for hver UC. Grænseflader lavet. BDD færdig. IBD udeladt
1.2	27-04-2015	SSK og MCM	Påbegyndt UML-klassediagram
2.0	29-04-2015	LSB, MFJ og AJF	Tilrettet domæne-, sekvens- og klassediagrammer, så design var klar til deadline
2.1	12-05-2015	LSB, MFJ og AJF	Rettet design-dokument i forhold til kommentar fra vejleder

3.1 Indledning

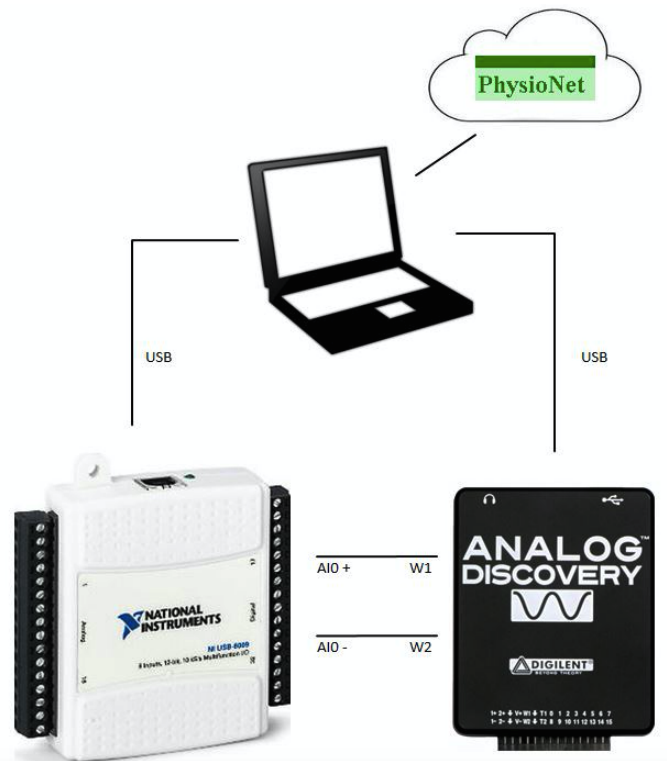
Formålet med design afsnittet er at beskrive hardwaren og softwaren for EKG-systemet. Det beskrives vha. diagrammer og skitser som tydeliggør, hvordan de forskellige klassers funktionalitet er, og hvordan klasserne snakker sammen.

3.2 Hardware arkitektur

I dette projekt er hardwaren blevet udleveret, derfor vil hardware arkitekturen ikke blive beskrevet ned i mindste detalje, men derimod kun hardwarens overordnede funktion. Hardwaren for systemet består af en National Instruments DAQ og en signalgenerator fra Analog Discovery. Disse er begge forbundet til en computer via USB-porte.

I dette projekt bliver EKG-signalet ikke målt fysisk via en patient, men simuleret via et signal, som er hentet ned fra websiden PhysioNet ¹.

¹<http://www.physionet.org>



Figur 3.1: Grafisk illustration af hardware opsætning

Som det ses på den overstående figur, er denne opstilling meget simpel. Figur 2.1 viser, hvordan de forskellige komponenter relaterer til hinanden. Display hardwaren vil i dette projekt være en computer, hvilket er et krav fra softwarens side, således at der kan vises et EKG-signal i form af en graf.

Som beskrevet ovenfor er det Analog Discovery, som sammen med EKG-signalets informationer fra PhysioNet, skaber vores fiktive patient. Før EKG-signalet vises på computerens skærm, skal det igennem de forskellige komponenter i opstillingen. Analog Discovery modtager EKG-signalets information som en CSV-fil, denne fil omdanner Analog Discovery til et analogt signal, som herefter sendes videre til DAQ'en. I DAQ'en digitaliseres det analoge signal, og tilpasses, så det efterfølgende kan udskrives på computer skærmen, i form af en EKG graf.

3.2.1 Grænseflader

Grænseflader af forbindelserne imellem de forskellige dele af hardwaren.

Forbindelse	Signaltype	Funktionalitet
DAQ - Computer	Digital	DAQ'en konverterer det analoge signal til digitalt og videresender det til computeren. Informationen sendes begge veje.
Computer - Analog Discovery	Digital	Computeren simulerer et EKG-signal og sender det til Analog Discovery.
Analog Discovery - DAQ	Analog	Analog Discovery konverterer signalet fra digitalt til analogt og videresender det til DAQ'en.

Tabel 3.2: Beskrivelse af grænseflader.

3.3 Software arkitektur

I dette projekt arbejdes der med objektorienteret programmering i programmeringssproget C#. Projektet er opbygget i henhold til trelagsmodellen. Præsentationslaget består af 4 GUI'er, logiklaget af en enkelt klasse, og datalaget består af 3 klasser, hvoraf den ene er en blackbox.

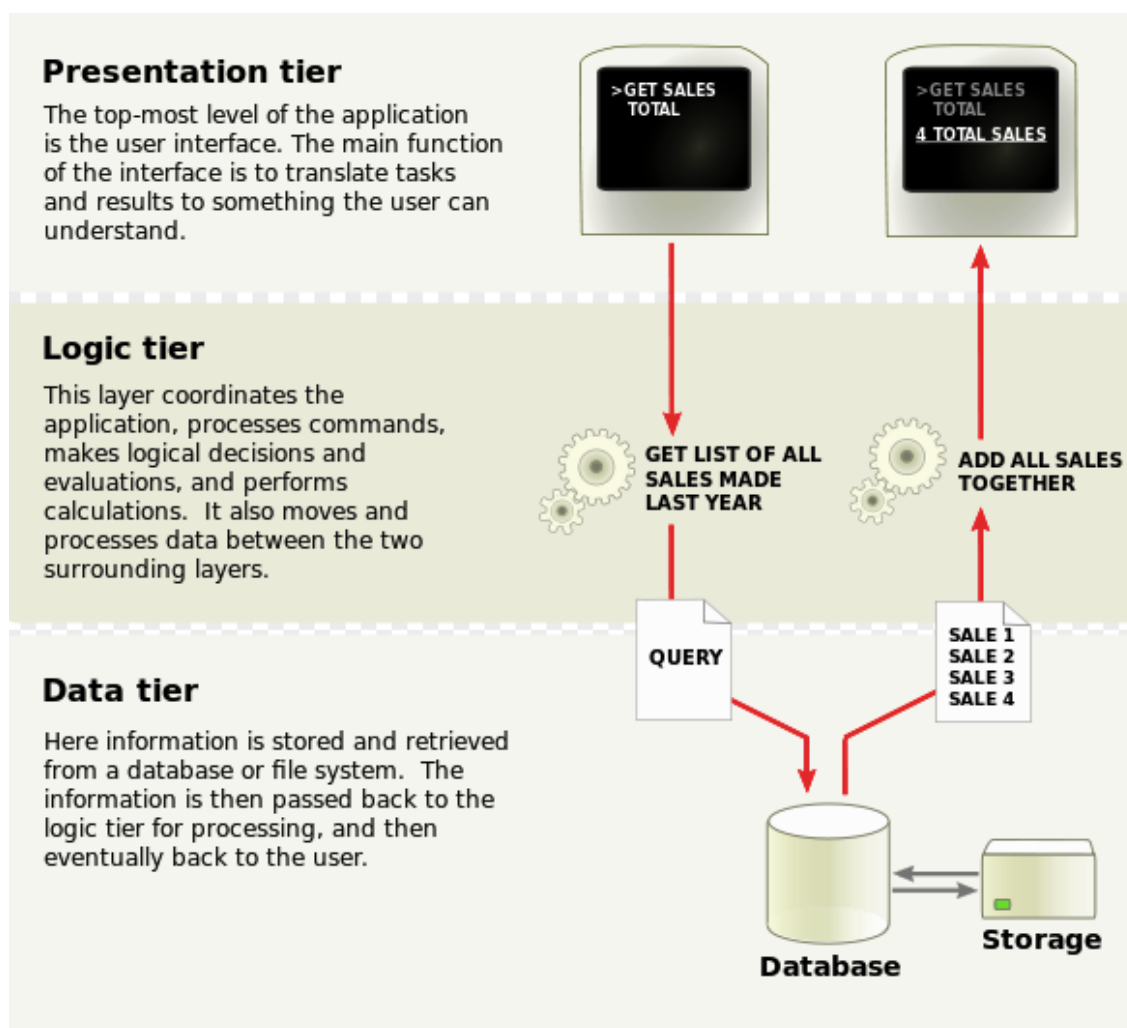
Trelagsmodellen

Trelagsmodellen er en model, som kan bruges til at opbygge et system. Idéen er, at systemet opdeles i mindre moduler/lag - et præsentationslag (grænseflade), et logiklag (funktionalitet) og et datalag (database), som spiller sammen.

Præsentationslaget er det øverste lag. Det er det lag, brugeren har indvirkning på og hvor de behandlede data præsenteres brugervenligt.

Logiklaget er det midterste lag. Dette lag er en slags bindeled mellem præsentationslaget og datalaget. I laget modtages og behandles informationerne fra præsentationslaget, og sendes videre til datalaget. Logiklaget kontrollerer hele systemets funktionalitet.

Datalaget er det nederste lag. Laget modtager dataerne fra logiklaget, som håndteres og lagres. Datalaget fungerer som en database, hvor man kan gemme nye data og hente gamle data frem igen. Trelagsmodellen vil typisk bygges grafisk op som vist på nedenstående figur 3.2.

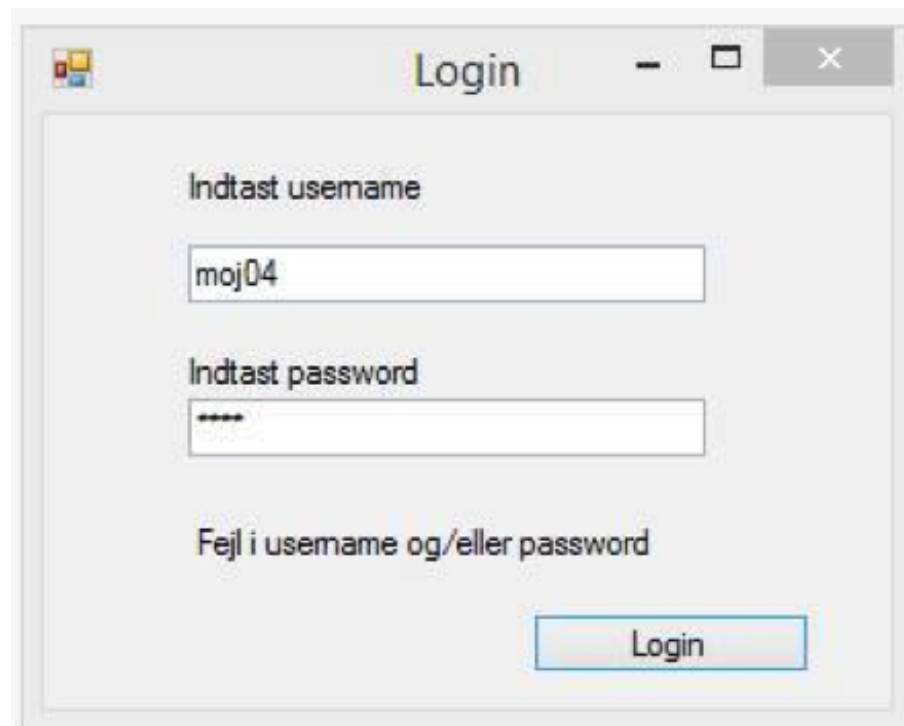
Figur 3.2: Trelagsmodel²

Altså fungerer et system, opbygget efter trelagsmodellen, således, at alle lag er uafhængige af hinanden. Denne opbygning medfører, at det systemet bliver mere overskueligt, samt det er muligt at forstå de enkelte lag hver for sig. At lagene er uafhængige er også en fordel når der skal ændres i systemet. Det gør det muligt at begrænse vedligeholdelse til det aktuelle lag. Desuden er det muligt, at modtage et lag fra et andet system, og ligeledes at sende og genbruge et lag andetsteds. I en projektgruppe er det også en fordel, at det er muligt at fordele arbejdet mellem gruppemedlemmer, og dermed kunne arbejde uafhængigt af andre.

3.3.1 GUI

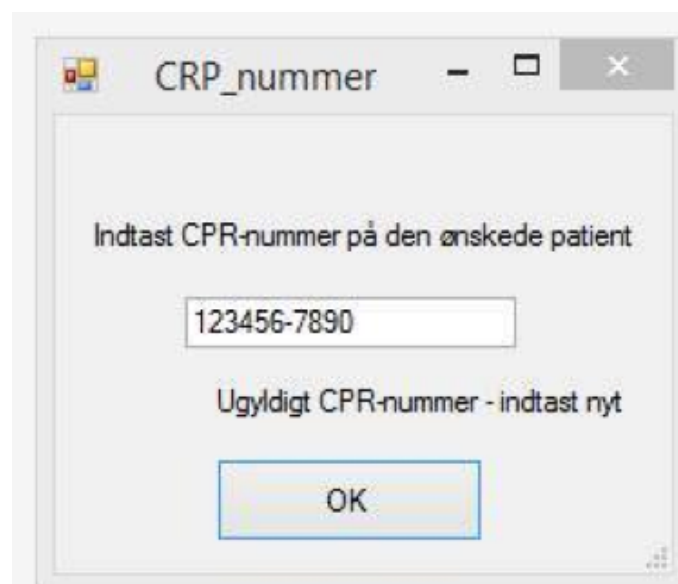
I dette afsnit vil GUI'erne beskrives nærmere. Præsentationslaget består i alt af 4 GUI'er eller vinduer - Login-vindue, CPR-vindue, EKG-vindue og Gem_måling-vindue.

²http://en.wikipedia.org/wiki/Multitier_architecture



Figur 3.3: Login-vindue

Når programmet startes op, vil et Login-vindue fremkomme på skærmen, se figur 3.3. Her skal brugeren logge ind for at få adgang til programmet, ved at indtaste sit personlige username og password. Bliver enten username eller password ikke godkendt, returneres en fejlmeddelelse, i form af en linje tekst nedenfor login-felterne. Bliver login godkendt åbnes CPR-vinduet, se figur 3.4.

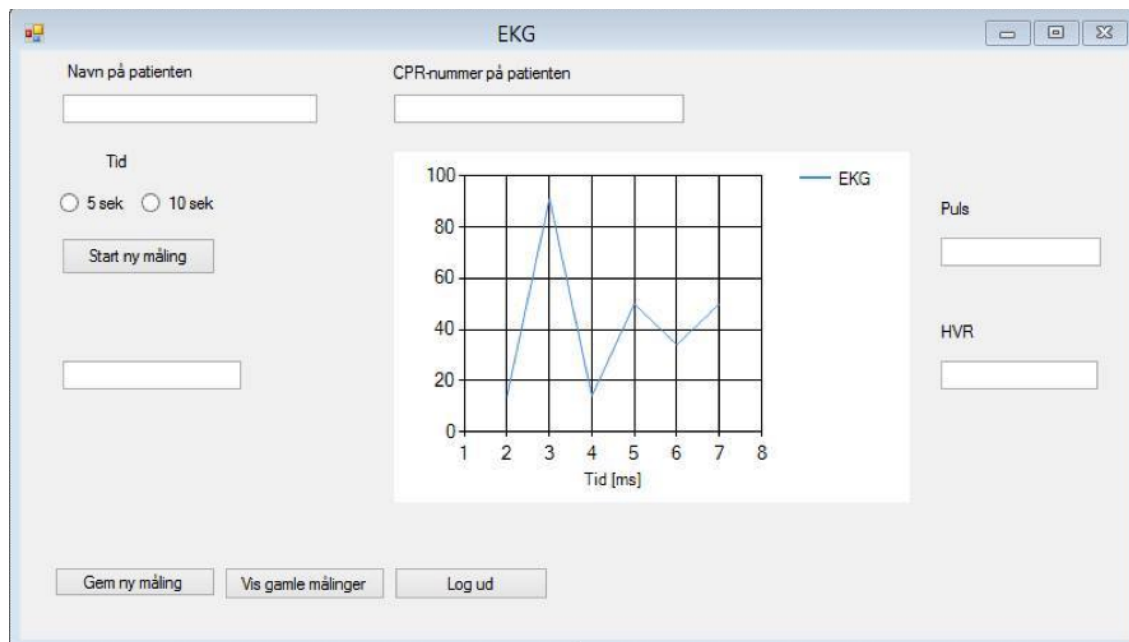


Figur 3.4: CPR-vindue

Her bliver brugeren bedt om at indskrive CPR-nummeret på den fiktive patient. CPR-

nummeret bruges til at identificere personen i forbindelse med fremtidige målinger, samt til gemme målingerne i hvad der svarer til den fiktive persons journal. Hvis CPR-nummeret ikke bliver godkendt, vil brugerne blive gjort opmærksom på dette via en fejlmelding.

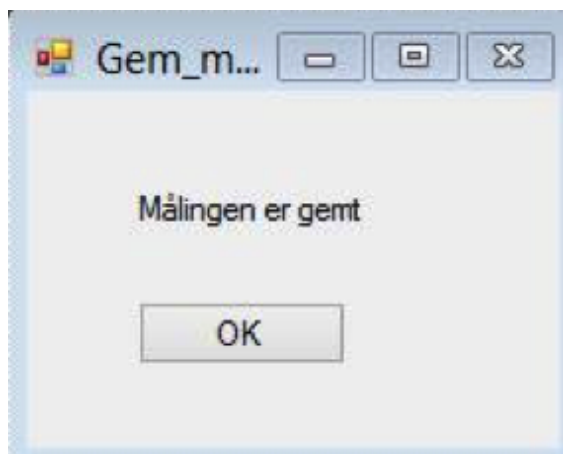
Når CPR-nummeret bliver godkendt, åbner det primære vindue, EKG-vinduet, se figur 3.5.



Figur 3.5: EKG-vindue

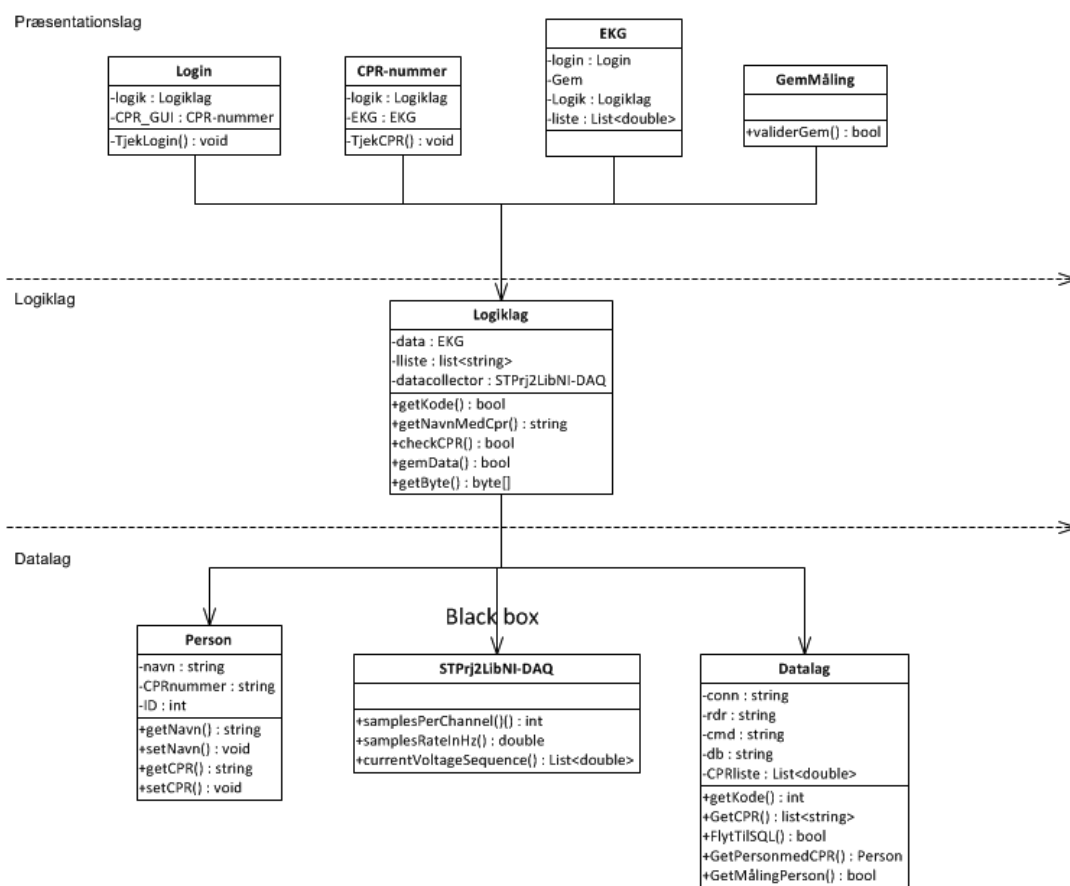
EKG-vinduet er der, hvor man kan starte og se målinger. Vinduet er udstyret med knapper, som angiver hvor lang tid brugeren ønsker en måling skal køre over. Til højre i vinduet vises grafen over EKG-signalet, kort efter der er trykket "Start ny måling". Yderligere til højre kan der i tekstboksen, ses puls for den pågældende graf. I det tomme tekstfelt, til venstre under knappen "Start ny måling", vil diagnosen for EKG-signalet udskrives. Nederst i vinduet er der knapper til at gemme målingerne, samt at logge ud fra programmet. Hvis man logger ud, kommer man tilbage til Login-vinduet, og EKG-vinduet er ikke længere synligt.

Det sidste vindue der er valgbart, frembringes når der trykkes på "Gem ny måling". Herefter vises pop-up vindue, som bekræfter, at den pågældende EKG-målingen er gemt, se figur 3.6.



Figur 3.6: Gem_måling-vindue

3.3.2 UML klassediagram



Figur 3.7: UML-klassediagram over softwaren

Det ovenstående UML diagram (figur 3.7) over projektets software, giver en bedre og lettere forståelse for selve koden. Her er det meget tydeligt, at koden er opbygget efter trelagsmodellen. Præsentationslaget er bygget op af de førnævnte GUI'er, som bliver interfacen for selve bruger-interaktionen. I dette lag sker der som udgangspunkt intet behandlingsar-

bejde. Opsætningen og funktionerne af grafiklaget er yderligere beskrevet under afsnittet 'GUI'.

Logiklaget består af en enkelt klasse, der fungerer som programmets primære styring. Logiklaget er til for, at grafiklaget ikke har direkte adgang til datalaget. Det er altså i dette lag, hvor alt programmets data bliver behandlet. Her bliver EKG'et f.eks. analyseret og login, samt CPR, bliver valideret i forhold til det data, der befinder sig i datalaget.

Datalaget består af to klasser, og en tilhørende blackbox. I det overstående er der kun beskrevet de metoder i blackboxen, som bliver brugt i andre klasser i programmet. I dette lag bliver der primært sørget for forbindelsen til databasen. Klassen "Datalag" er kernen i dette lag. Her bliver der sørget for at gemme og hente informationer fra databasen. Det er i dette lag, hvor dataen som logiklaget arbejder med, kommer fra. Klassen "Person", er til for at binde alle person-informationer sammen, som medfører at det kun er nødvendigt at kalde et enkelt objekt, frem for 4 attributter.

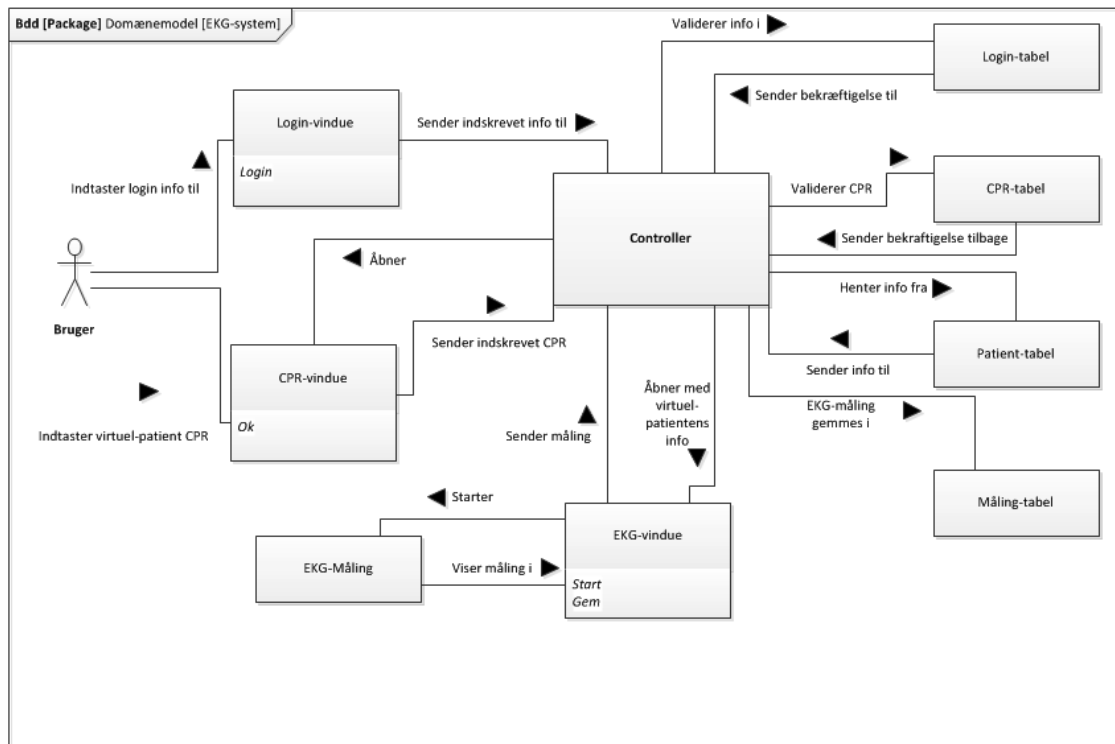
3.3.3 Applikationsmodel

En standard applikationsmodel vil i almindelig forstand indeholde en overordnet domænemodel, herefter klassediagram samt sekvensdiagram og tilsidst et opdateret klassediagram, hvor metoderne fra sekvensdiagram er inkluderet.

Applikationsmodellen for dette projekt består af en overordnet domænemodel, herefter sekvensdiagram for de forskellige Use Cases med tilhørende opdateret klassediagram, som beskriver metodekald og kommunikation mellem klasserne. Klassediagrammerne er her undladt, da de er irrelevante for dokumentationen af projektet. Applikationsmodellen er udarbejdet ud fra Use Cases, hvilket medfører, at metoderne er fiktive, altså ikke hentet direkte fra softwaren.

Domænemodel

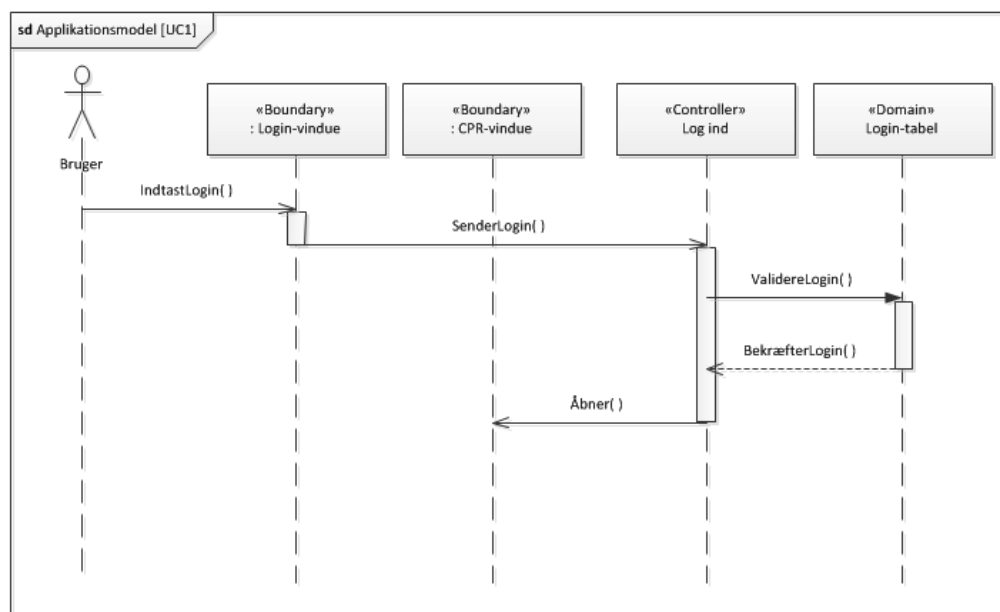
Domænemodellen er skabt på baggrund af de fem Use Cases. Gennem navneordsanalyse af Use Casene er de konceptuelle klasser fundet. I modellen beskrives, hvordan de konceptuelle klasser interagerer med hinanden. Controlleren er ikke en konceptuelle klasse, men er den, der sørger for at systemet fungerer optimalt.



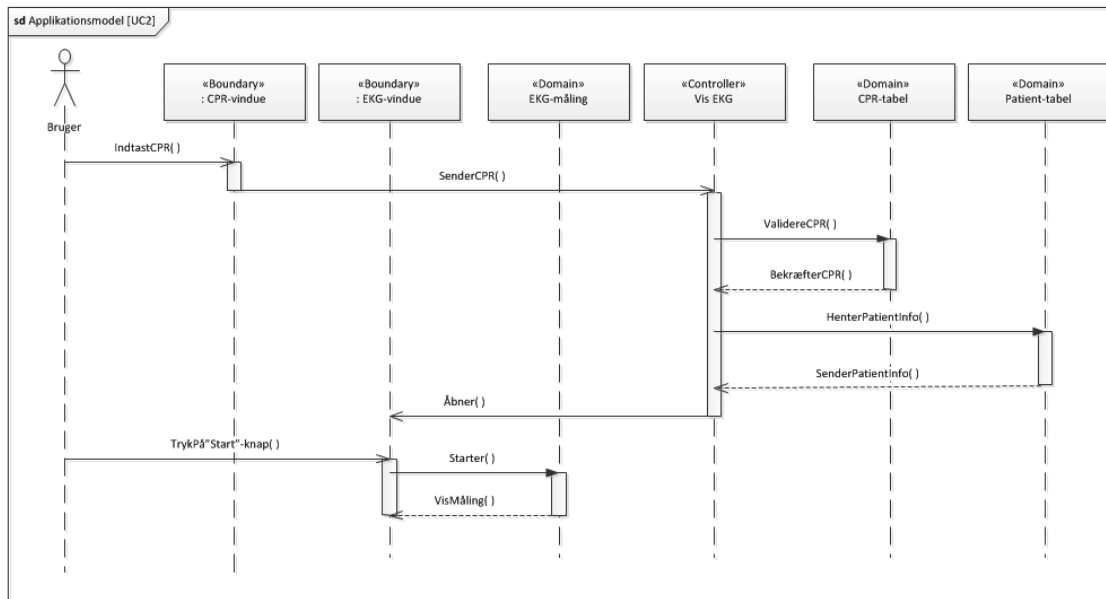
Figur 3.8: Domænemodel af EKG-systemet

Sekvensdiagram

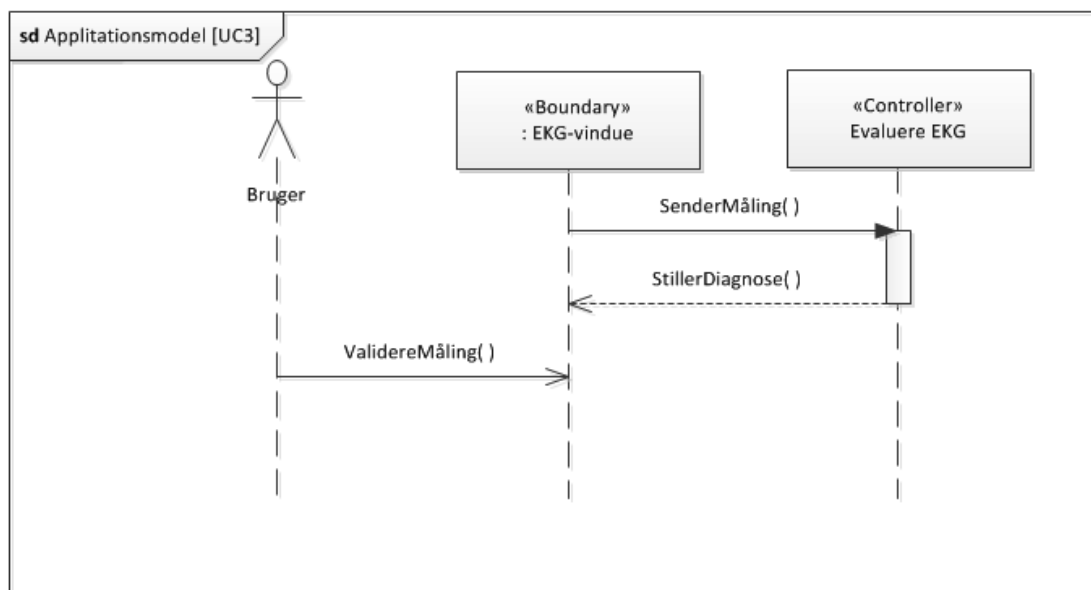
Sekvensdiagrammerne beskriver step-by-step, via fiktive metoder, forløbet i de forskellige Use Cases. Der er lavet et sekvensdiagram for hver Use Case, for at gøre systemet mere overskueligt. Et sekvensdiagram består af boundary-klasserne og domain-klasserne fra domænemodellen, samt en controller-klasse, med navn efter den specifikke Use Case.



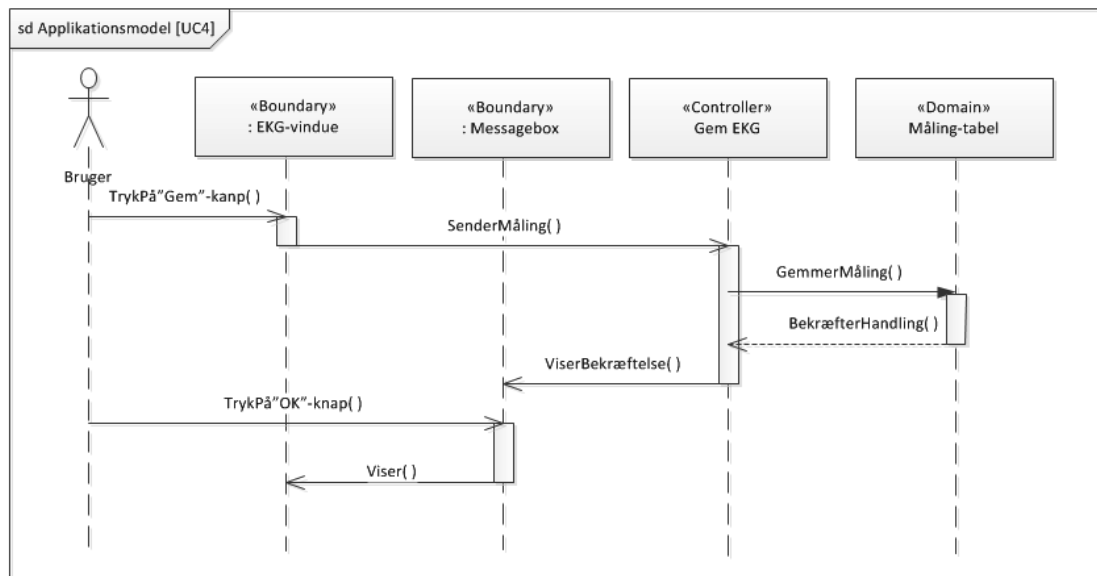
Figur 3.9: Sekvensdiagram for UC1



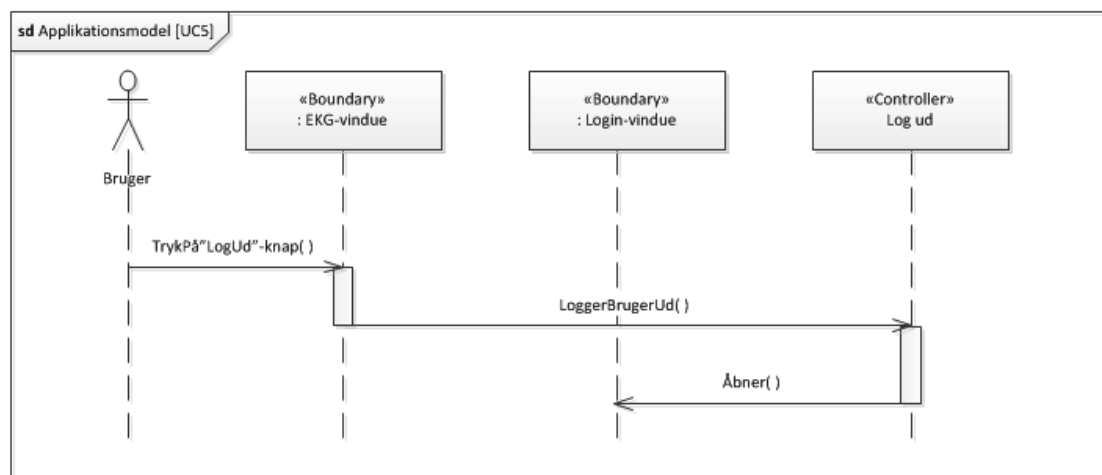
Figur 3.10: Sekvensdiagram for UC2



Figur 3.11: Sekvensdiagram for UC3



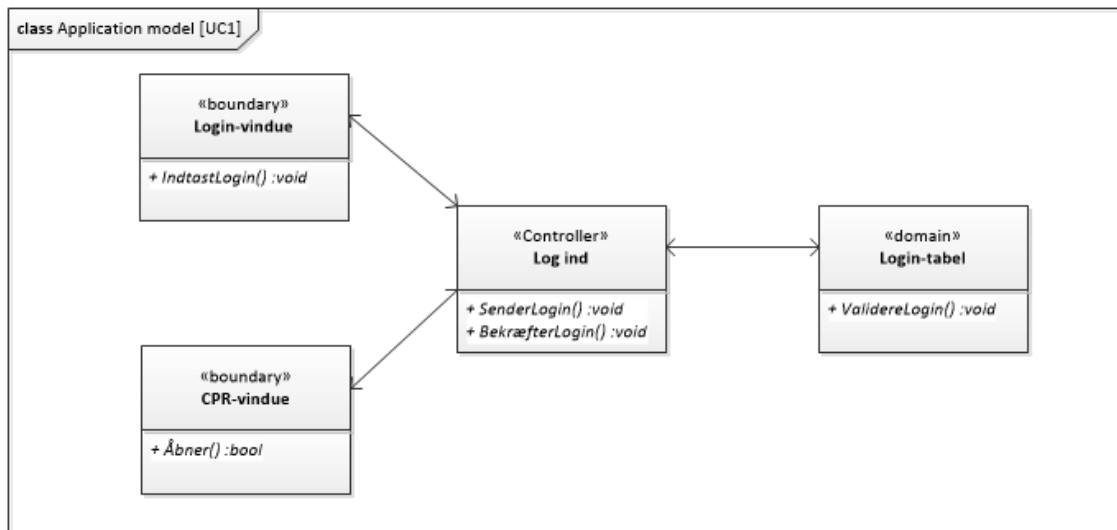
Figur 3.12: Sekvensdiagram for UC4



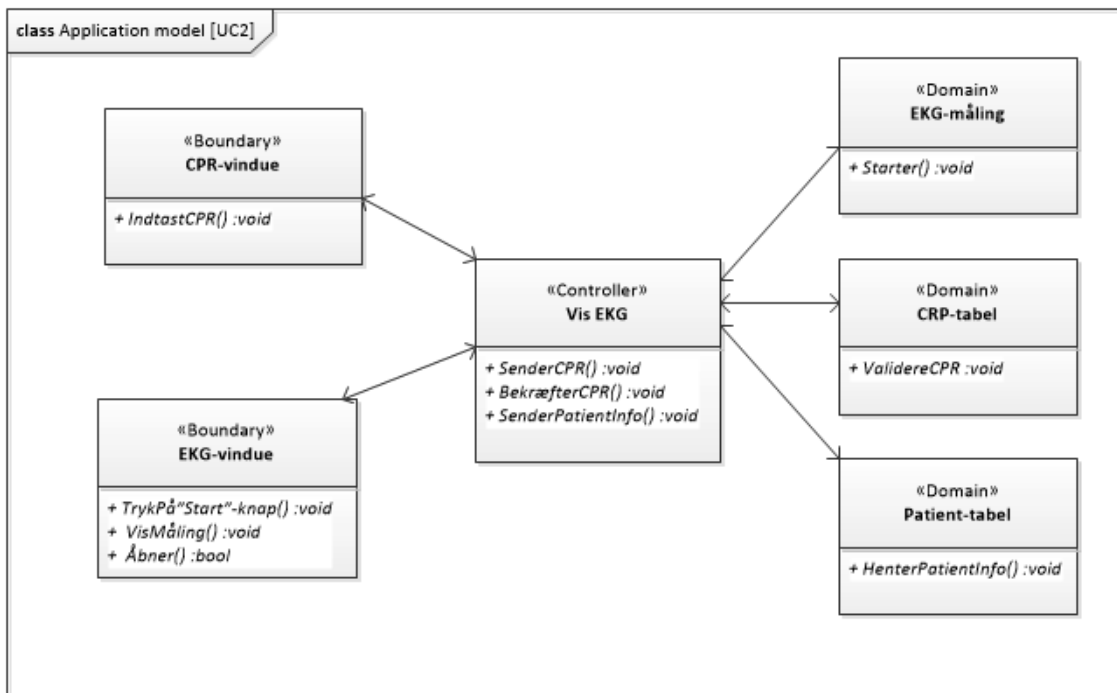
Figur 3.13: Sekvensdiagram for UC5

Opdateret Klassediagram

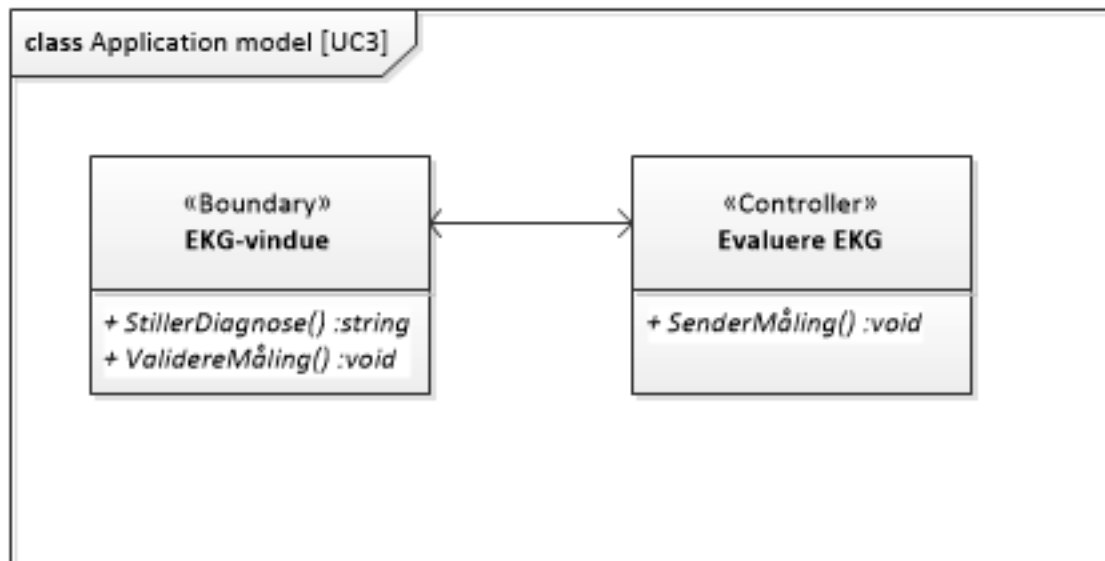
De opdateret klassediagrammer indeholder metoderne fra de dertilhørende sekvensdiagrammer - dette giver et overblik over, hvilke metoder de forskellige klasser består af.



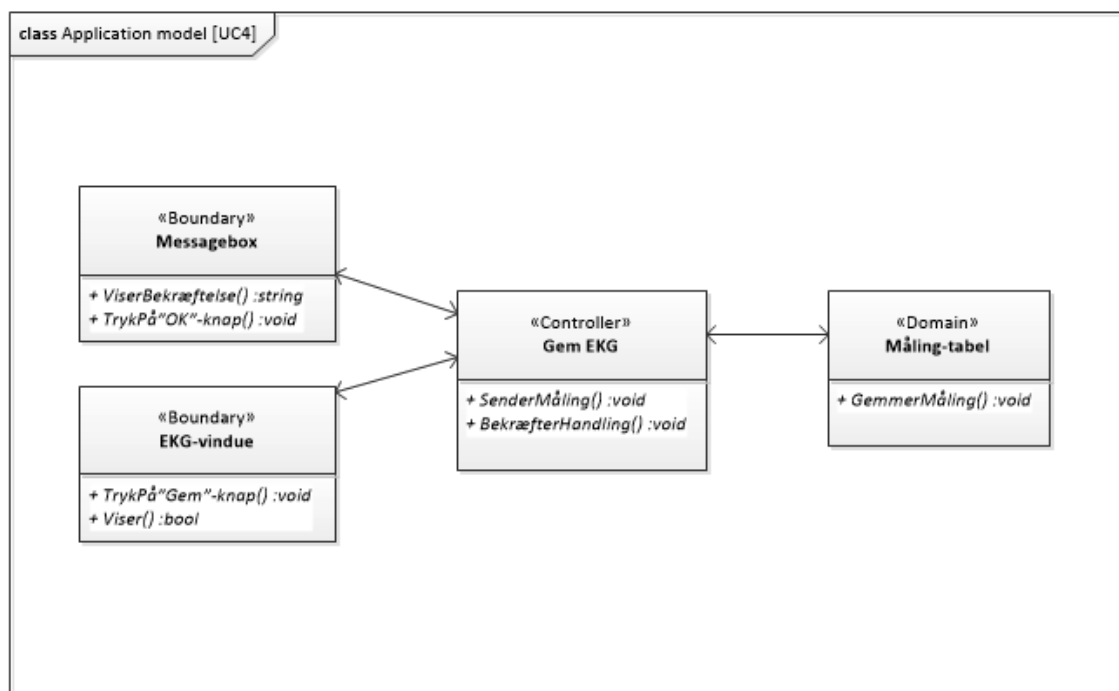
Figur 3.14: Klassediagram for UC1



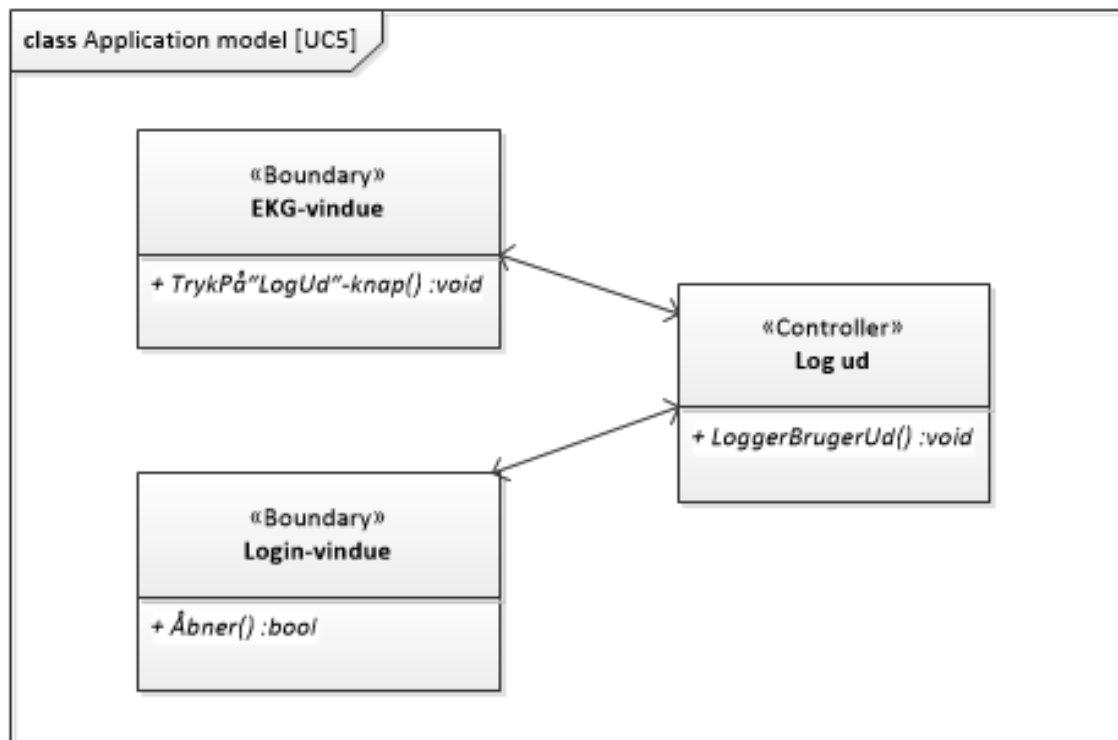
Figur 3.15: Klassediagram for UC2



Figur 3.16: Klassediagram for UC3



Figur 3.17: Klassediagram for UC4



Figur 3.18: Klassediagram for UC5

Accepttest 4

Version	Dato	Ansvarlig	Beskrivelse
1.0	18-03-2015	LSB, AJF og MFJ	Påbegyndt tilrettelse i forhold til den valgte sygdom, Atrieflimren.
1.1	26-03-2015	LSB, AJF, MFJ og CAA	AT færdigskrevet og klar til review
2.0	09-04-2015	LSB, AJF, MFJ og CAA	Rettet i forhold til review-kommentarer
Tekst	Tekst	Tekst	Tekst.

4.1 Accepttest af Use Cases

4.1.1 Use Case 1

Log ind

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenario</i>			
1. Indtast username samt password	Username- og passwordboks bliver udfyldt		
2. Tryk på "Login-knappen.	Login bliver godkendt. Login-vinduet lukkes ned mens CPR-vinduet åbnes		
<i>Exentions</i>			
2a. Username eller password er forkert	Besked vises på skærmen med tekst, der informerer om, at brugernavn eller password er forkert		

Tabel 4.2: Accepttest af Use Case 1.

4.1.2 Use Case 2

Vis EKG

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Indtast virtuel patients CPR-nummer	CPR-nummerboks bliver udfyldt		
2. Vælg indstillinger	Indstillinger bliver valgt		
3. Tryk på "Start ny måling"	Målingen startes i EKG-vinduet		
4. EKG-data illustreres på en graf	En analyserebar graf fremvises i EKG-vinduet		
1.a CPR-nummeret findes ikke. Besked vises med tekst, der informerer om, at CPR-nummeret ikke er gyldigt	Nyt CPR-nummer indtastes		
2.a Ingen ændring i indstillinger	Målingen foretages med default-indstillingerne		

Tabel 4.3: Accepttest af Use Case 2.

4.1.3 Use Case 3

Evaluer EKG

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Validere programets analyse af EKG-signalet	Det er muligt at se små fluktuationer, som kan aflæses på EKG-grafen		

- | | | |
|----|-----------------------------|--|
| 2. | Stil diagnosen atrieflimmer | Atrieflimmer kan aflæses ud fra EKG-grafen |
|----|-----------------------------|--|

Exentions

- | | | |
|-----|---|--|
| 2a. | Atriefrekvensen er ikke i intervallet 220-300 pr. minut | Det er ikke muligt at diagnosticere atrieflimmer ud fra EKG-grafen |
|-----|---|--|
-

Tabel 4.4: Accepttest af Use Case 3.

4.1.4 Use Case 4

Gem EKG

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Tryk på "Gem-knappen.	Messagebox kommer frem med besked om at målingen er gemt		
2. Tryk på "Ok-knappen	Målingen er gemt, vinduet lukkes og EKG-vinduet vises igen		
<i>Exentions</i>			

Tabel 4.5: Accepttest af Use Case 4.

4.1.5 Use Case 5

Log ud

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedscenarie</i>			
1. Tryk på "log ud-knappen	EKG-vinduet lukkes ned, mens login-vinduet fremkommer		

Exentions

Tabel 4.6: Accepttest af Use Case 5.

4.2 Accepttest af ikke-funktionelle krav

Ikke-funktionelt krav	Test/handling	Forventet resultat	Faktiske observationer	Godkendt
<i>Usability</i>				
Brugeren skal kunne starte en default-måling maksimalt 20 sekunder efter opstart af program	Start programmet, hvorefter der vha. stopur måles opstartstiden	At programmet er startet op indenfor 20 sekunder		
Brugeren skal have mulighed for at ændre tidsintervallet før målingerne foretages	Start programmet og ændrer indstillingerne i toolbar	At der er mulighed for at ændre indstillinger		
Login-vinduet skal indeholde en login-knap til at logge på og få vist EKG-vinduet	login-knappen er synlig i GUI, og ved tryk på knappen vises EKG-vinduet	At EKG-vinduet vises		
EKG-vinduet skal indeholde en "start-knap" til at igangsætte målingerne	Startknappen er synlig i GUI, og ved tryk på knappen igangsættes måling	At målingen igangsættes		

EKG-vinduet skal indeholde en "stop-knap" til at afslutte målingerne	Stopknappen er synlig i GUI, og ved tryk på knappen afsluttes måling	At målingen afsluttes
EKG-vinduet skal indeholde en "gem-knap" til at gemme målingerne	Gem-knappen er synlig i GUI, og ved tryk på knappen gemmes måling i database	Messageboks vises på skærmen med teksten "Måling er gemt" og kan findes i databasen
EKG-vinduet skal indeholde en "log ud-knap" til at logge ud	"log ud-knappen er synlig i GUI, og ved tryk på knap lukkes EKG-vinduet og login-vinduet vises	Login-vinduet vises
Målingen stopper automatisk efter det valgte tidsinterval	Der vælges et tidsinterval. Måling startes	Målingen stopper efter det valgte tidsinterval
<i>Reliability</i>		
Systemet skal have en effektiv MTBF på 20 minutter og MTTR på 1 minut	Køre programmet i 20 minutter. Genstart derefter programmet, hvor der tages tid med et stopur	Programmet har kørt i 20 minutter og genstartes indenfor 1 minut
<i>Performance</i>		
Der skal vises en EKG-graf i interfacet, hvor spænding vises op ad y-aksen (-1V til 1V) og tiden på x-aksen	Gennemfør en måling	At spændingen for EKG-signalet er op ad y-aksen, samt tiden hen ad x-aksen

Det skal være muligt at kunne scrolle igennem målingerne hen ad x-aksen	Der gennemføres en måling hvor efter der scrolles hen ad x-aksen	At der ved scrolling kan ses forskellige dele af EKG-signalet hen ad x-aksen
Der skal kunne tages et sample over et brugerbestemt interval, hvor frekvensen er tilpasses målingerne, således at grafen er analyserbar	Gennemfør en måling, hvor et bestemt interval hen ad x-aksen er valgt	At det valgte interval synliggøres
<i>Supportability</i>		
Software er opbygget af tre-lagsmodellen	Kig i koden efter data-lag, logik-lag og GUI-lag	At koden indeholder et data-lag, et logik-lag og et GUI-lag

Tabel 4.7: Accepttest af Ikke-funktionelle krav

