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

25. maj 2015

Gruppemedlemmer

Lise Skytte Brodersen (201407432)	Dato
Mads Fryland Jørgersen (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	Unifled modeling language

Indholdsfortegnelse

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 intergerer, 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 illusterret 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.

Kravspecifikation 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, Atrie-flimren
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

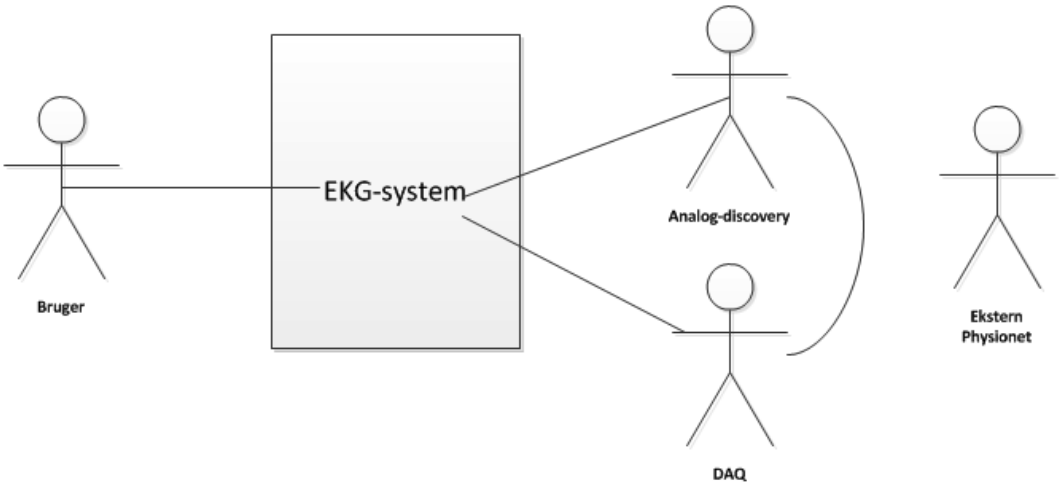
Kravspecifikationen vil beskrive, ud fra en række modeller, hvordan EKG-systemet fungerer. Helt generelt er EKG-måling en simpel metode, til at måle hjertets elektriske aktivitet via elektroder, som registrerer elektriske impulser, placeret på huden. Ud fra disse impulser dannes en graf, som benyttes 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. Via Analog-discovery omdannes csv-filens data til et analogt signal, som derefter omdannes til et digitalt signal i DAQ'en. Dette signal sendes til EKG-systemet, som ud fra disse data danner en graf. Programmet detekterer markørudsving i EKG-grafen, som derefter valideres og analyseres af brugeren.



Figur 2.1: Aktør-kontekstdiagram

2.2.2 Aktørbeskrivelse

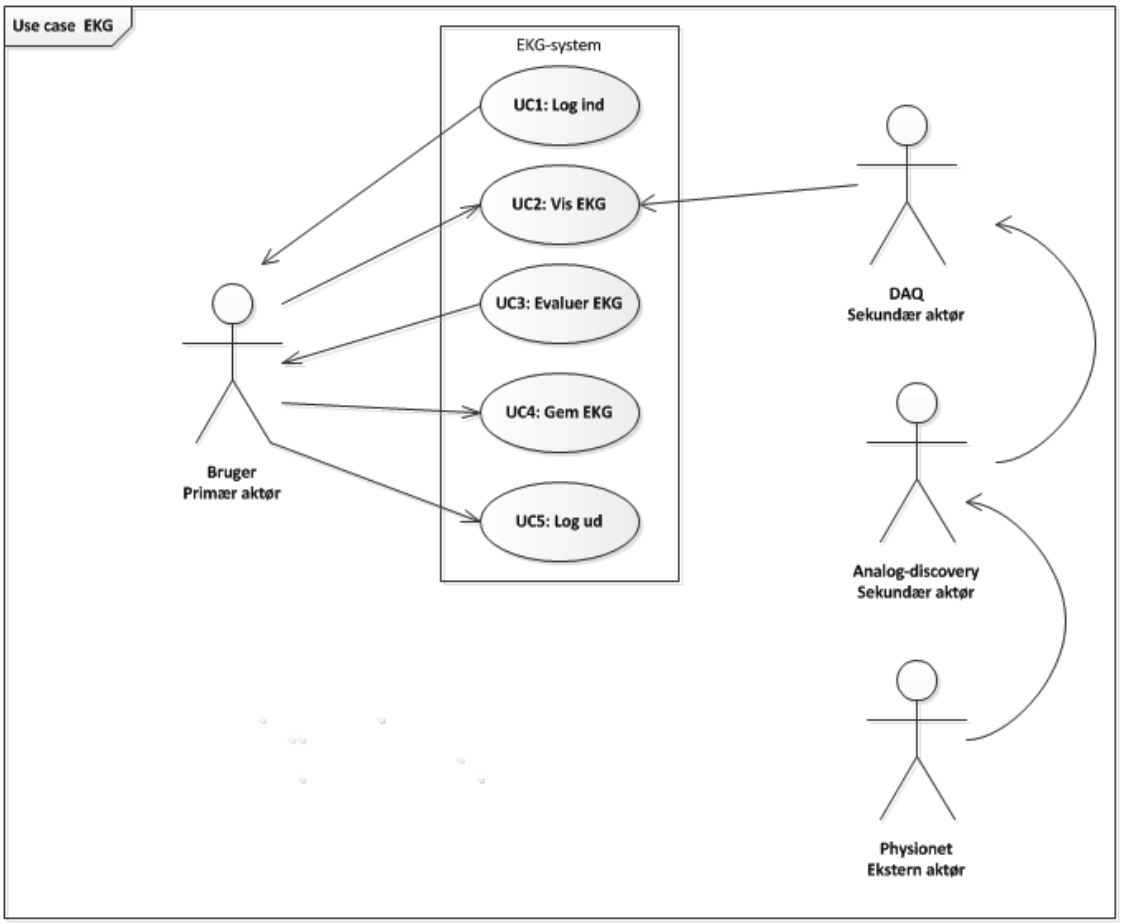
Aktørnavn	Type	Beskrivelse
Bruger	Primær	Brugeren er den aktør, der ønsker at foretage målingerne, som omfatter EKG samt diagnosticering af artieflimmer. Brugeren er en person, der har kendskab til EKG-systemet. Fx sundhedsfaglig personale
Analog-discovery	Sekundær	Analog-discovery omdanner data fra 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-discovery samt i DAQ'en, den 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	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 forsættes 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-discovery
Sekunær 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-discovery og DAQ'en er koblet til og data er hentet ned
Resultat		EKG-graf bliver vist
Hovedforløb	1.	Brugeren indtaster virtuel patients CPR-nummer
	2.	Brugeren trykker på "Ok"-knappen. CPR-vinduet lukkes ned, mens EKG-vinduet åbnes [2.a *CPR-nummeret findes ikke*]
	3.	Målingen startes ved at trykke på "Start-ny-måling"
	4.	EKG-data illustreres på en graf
Undtagelser	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

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	1.	Brugeren validere programmets analyse af EKG-signalet
	2.	Brugeren stiller diagnosen atrieflimmer [2.a *Atriefrekvensen er ikke i intervallet 220-300 pr. minut*]
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	1.	Brugeren trykker på "Gem-ny-måling"-knappen. EKG-målingerne gemmes på en lokal og offentlig database. 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) 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 "log ud"-knap
- (M) EKG-vinduet skal indeholde en "gem"-knap
- (M) Information-vinduet skal indeholde en "gem"-knap

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.

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

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

Supportability

- (M) Softwaren er opbygget af trelagsmodellen

Design 3

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

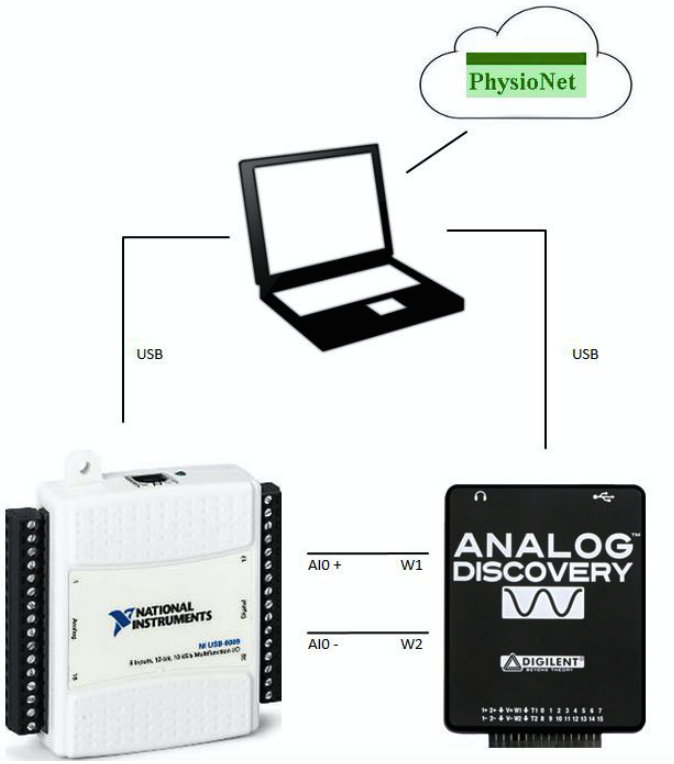
3.1 Indledning

Formålet med design afsnittet er at beskrive hardwaren og softwaren for EKG-systemet. Det beskrives vha. diagrammer og skitser som tydeligør, 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.

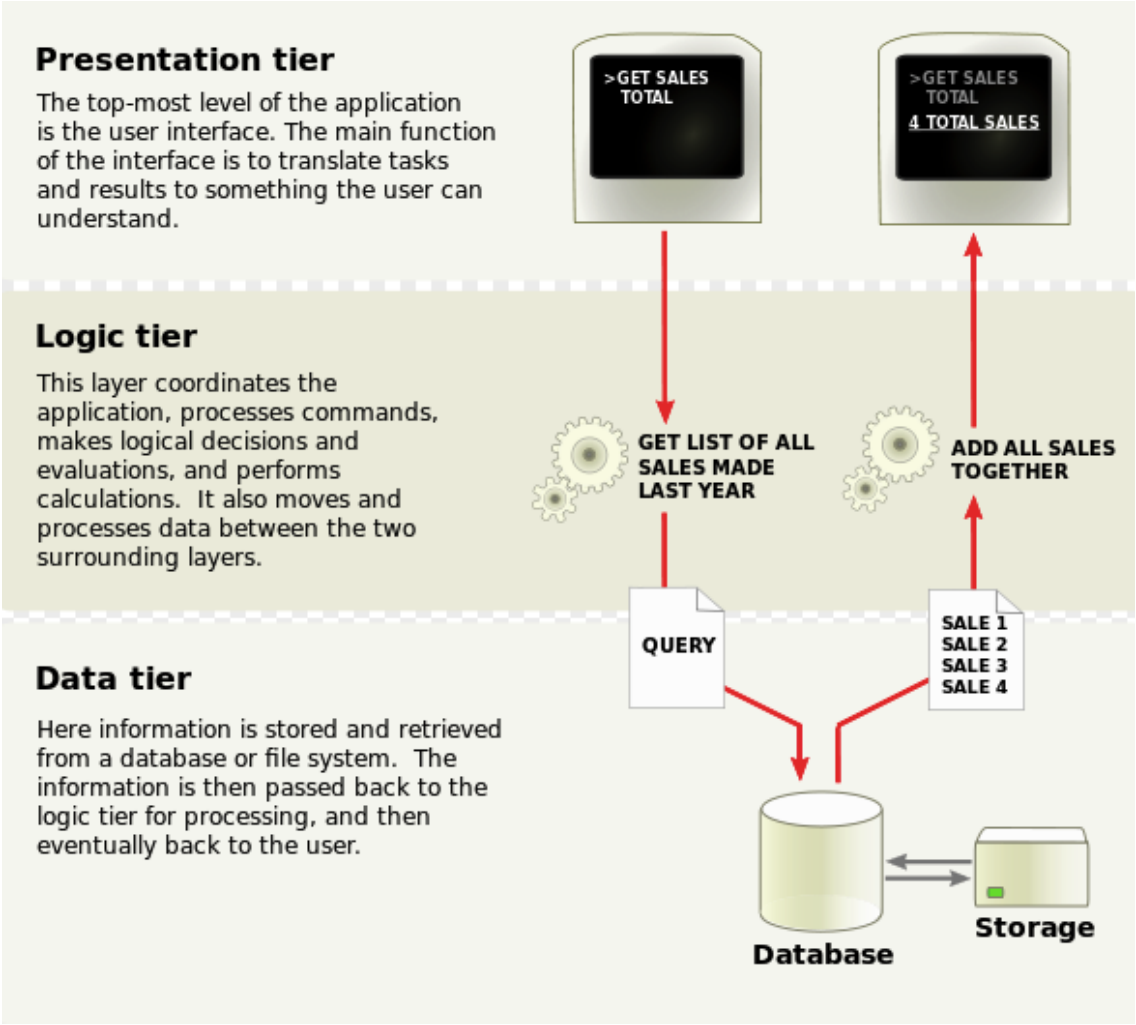
Tabel 3.2: Beskrivelse af grænseflader.

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 fungere som bindeled til databaser, 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.



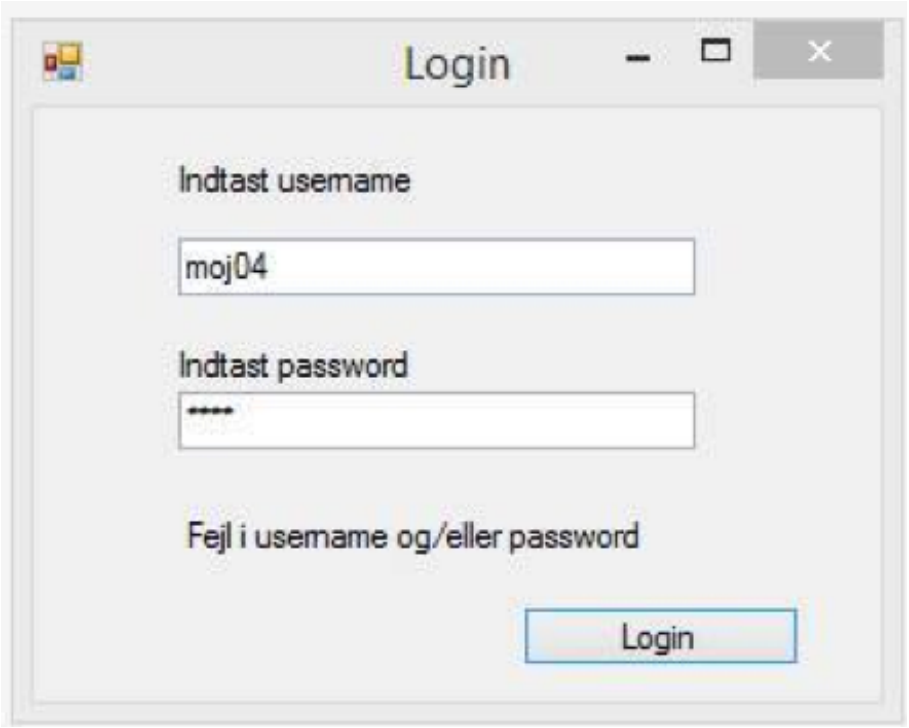
Figur 3.2: Trelagsmodel[2]

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

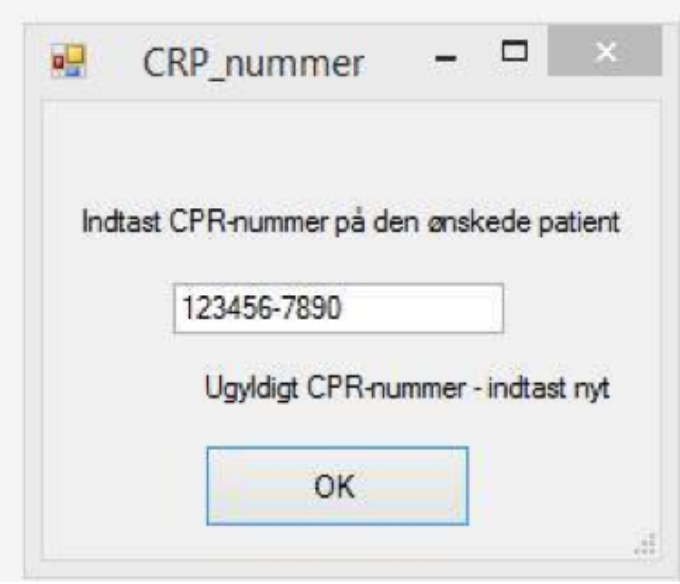
I dette afsnit vil GUI'erne beskrives nærmere. Præsentationslaget består i alt af 4 GUI'er eller vinduer - Login-vindue, CPR-vindue, EKG-vindue og Gem_måling-vindue.

[2] http://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.

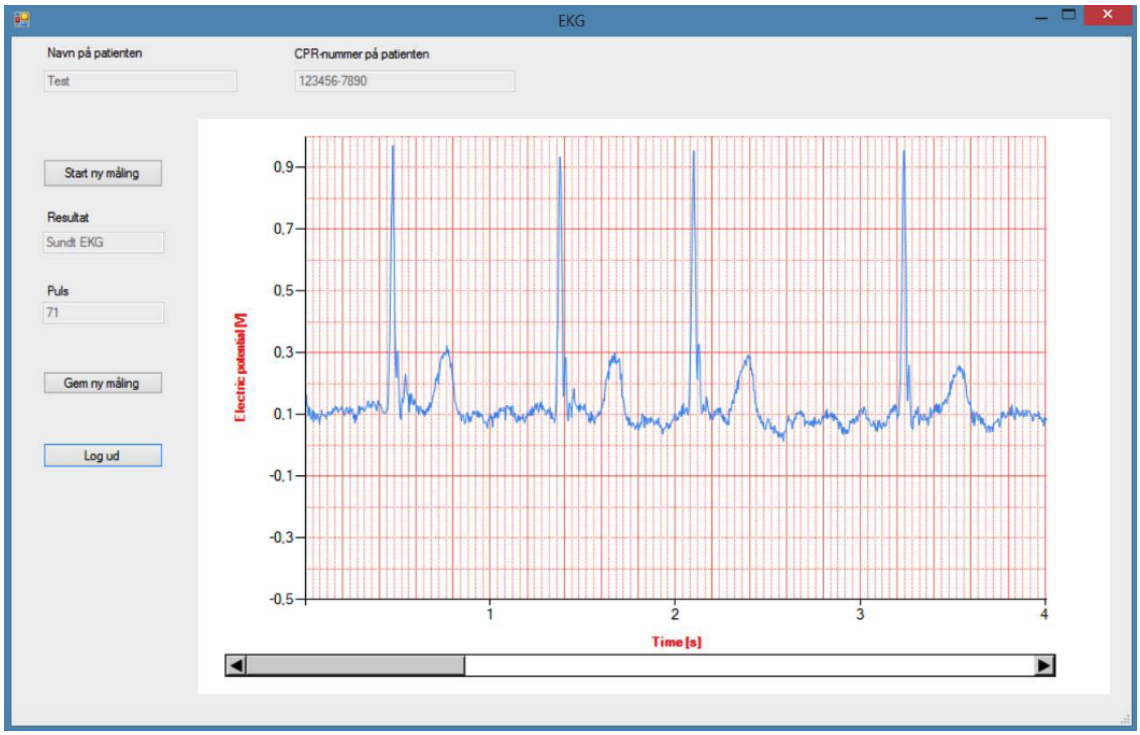


Figur 3.4: CPR-vindue

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.

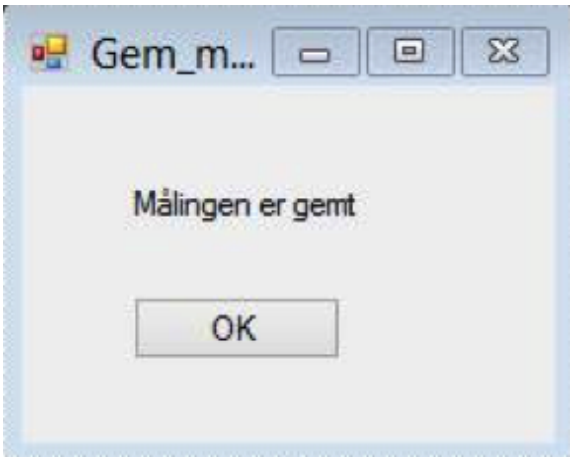
Når CPR-nummeret bliver godkendt, åbner det primære vindue, EKG-vinduet, se figur 3.5.



Figur 3.5: EKG-vindue

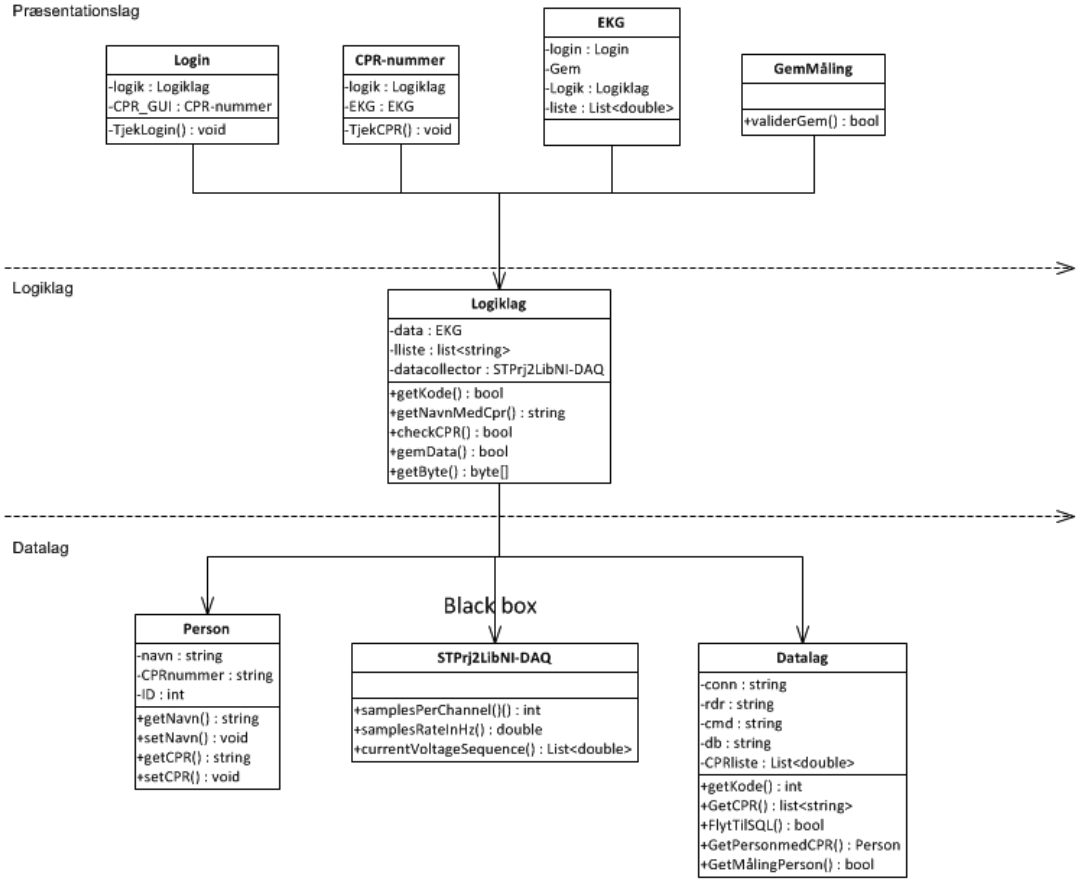
EKG-vinduet er der, hvor man kan starte og se målinger. Vinduet er udstyret med knapper, som angiver hvor lang tid brugeren ønsker en måling skal køre over. Til højre i vinduet vises grafen over EKG-signalet, kort efter der er trykket "Start ny måling". Yderligere til højre kan der i tekstboksen, ses 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 trelags-modellen. Præsentationslaget er bygget op af de førnævnte GUI'er, som bliver interfacen for selve bruger-interaktionen. I dette lag sker der som udgangspunkt intet behandlings-

arbejde. Opsætningen og funktionerne af grafiklaget er blevet 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.

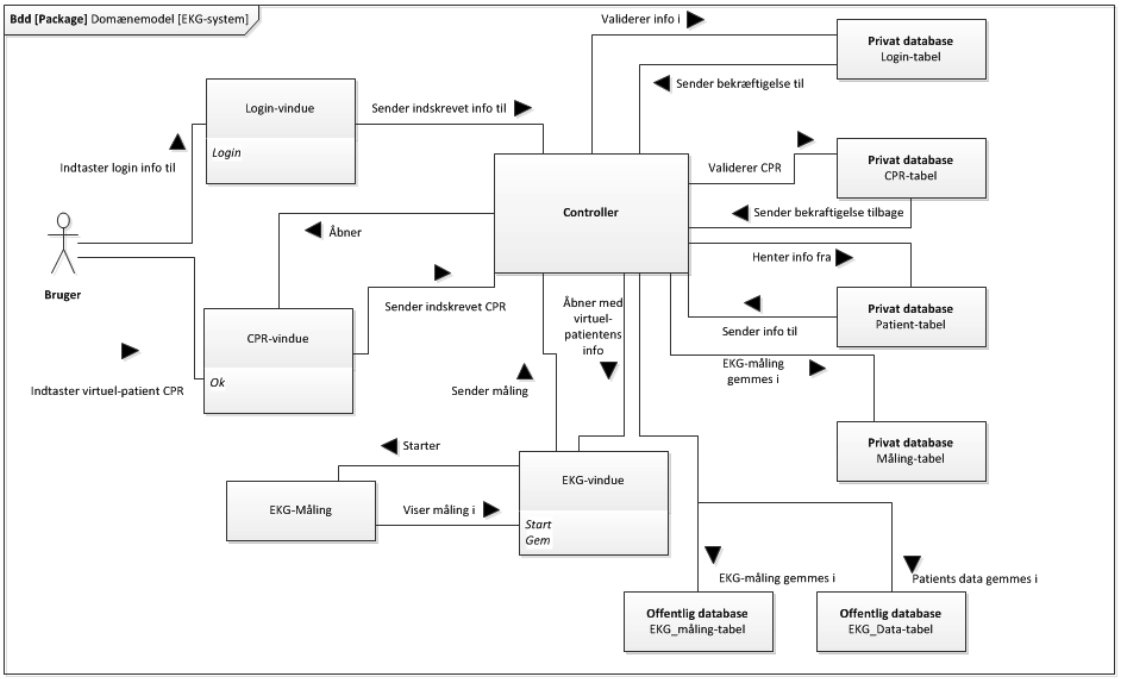
3.3.3 Appliktationsmodel

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

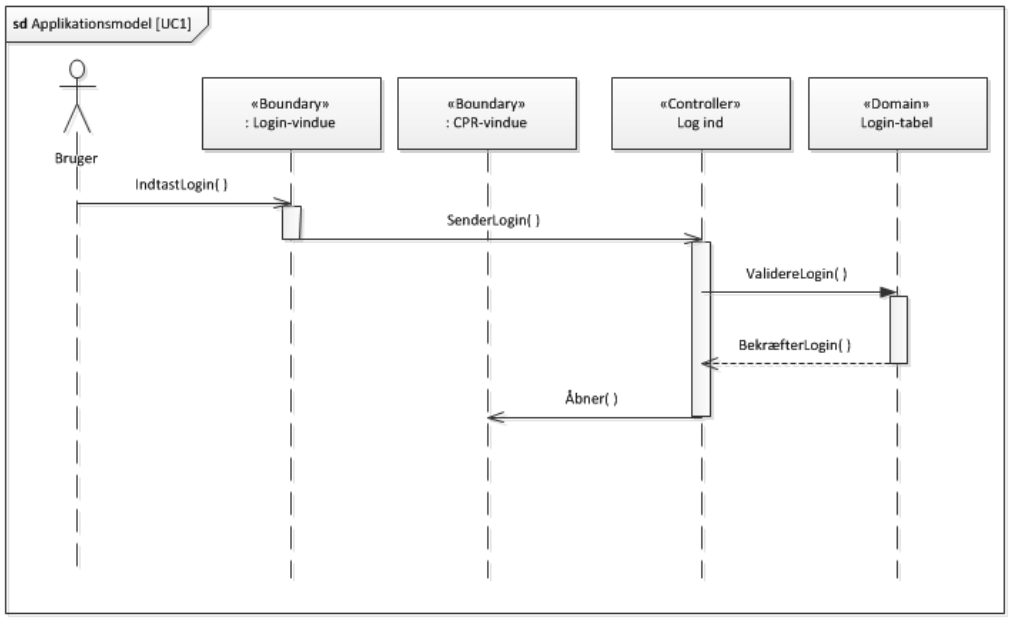
Domænemodellen er skabt på baggrund af de fem Use Cases. Gennem navneordsanalyse af Use Casene er de konceptuelle klasser fundet. I modellen beskrives, hvordan de konceptuelle klasser interagerer med hinanden. Controlleren er ikke en konceptuelle klasse, men er den, der sørger for at systemet fungerer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



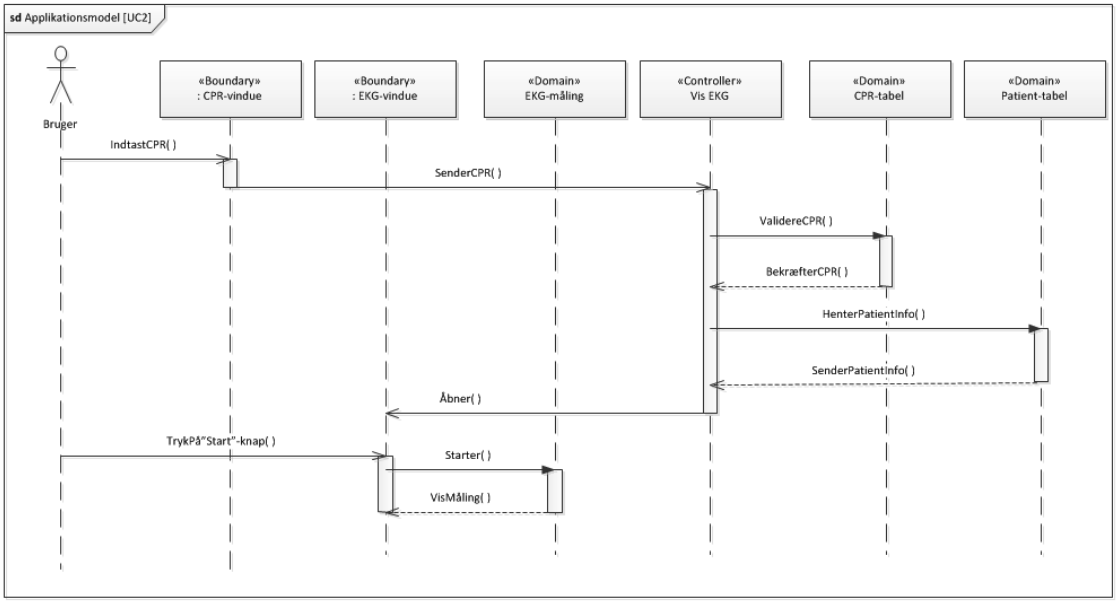
Figur 3.8: Domænemodel af EKG-systemet

Sekvensdiagram

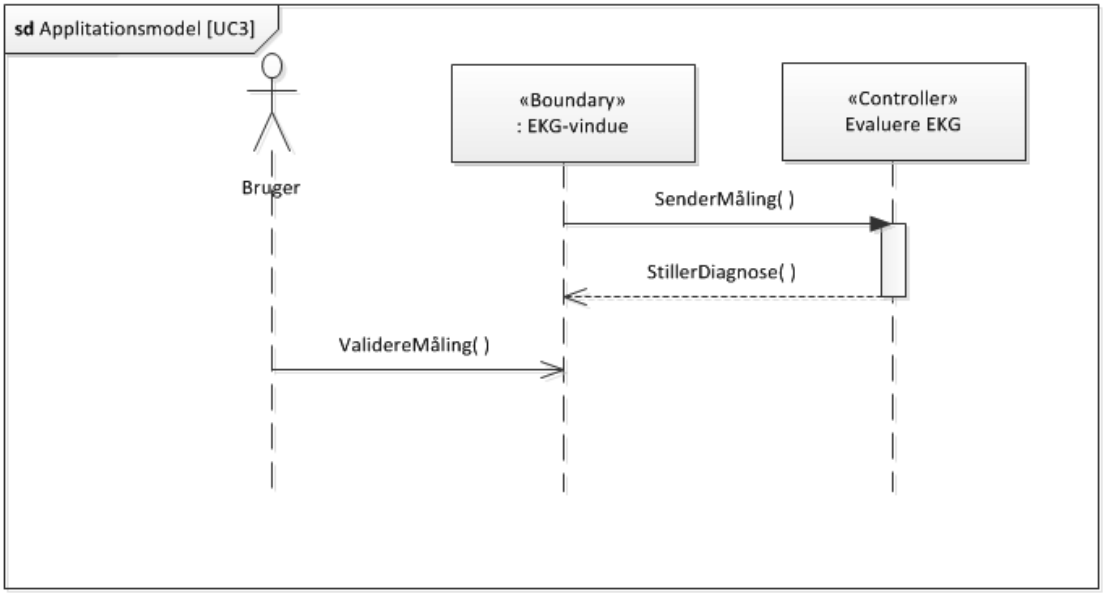
Sekvensdiagrammerne beskriver step-by-step, via fiktive metoder, forløbet i de forskellige Use Cases. Der er lavet et sekvensdiagram for hver Use Case, for at gøre systemet mere overskueligt. Et sekvensdiagram består af boundary-klasserne og domain-klasserne fra domænemodellen, samt en controller-klasse,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



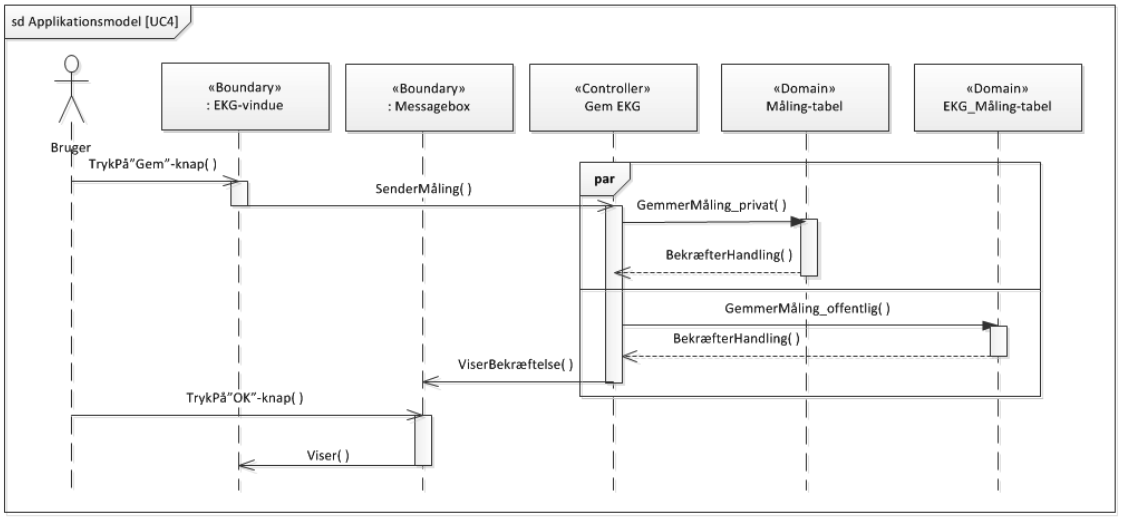
Figur 3.9: Sekvensdiagram for UC1



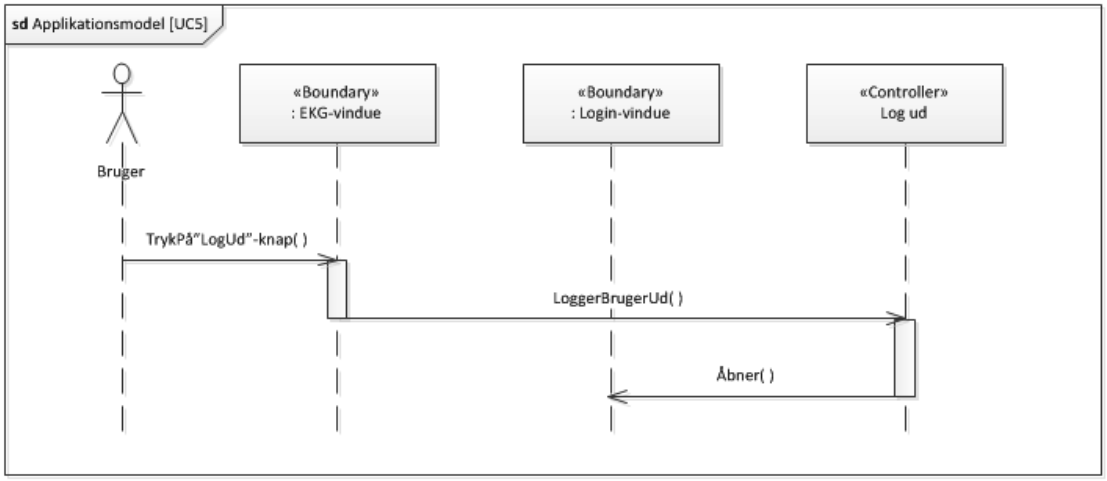
Figur 3.10: Sekvensdiagram for UC2



Figur 3.11: Sekvensdiagram for UC3



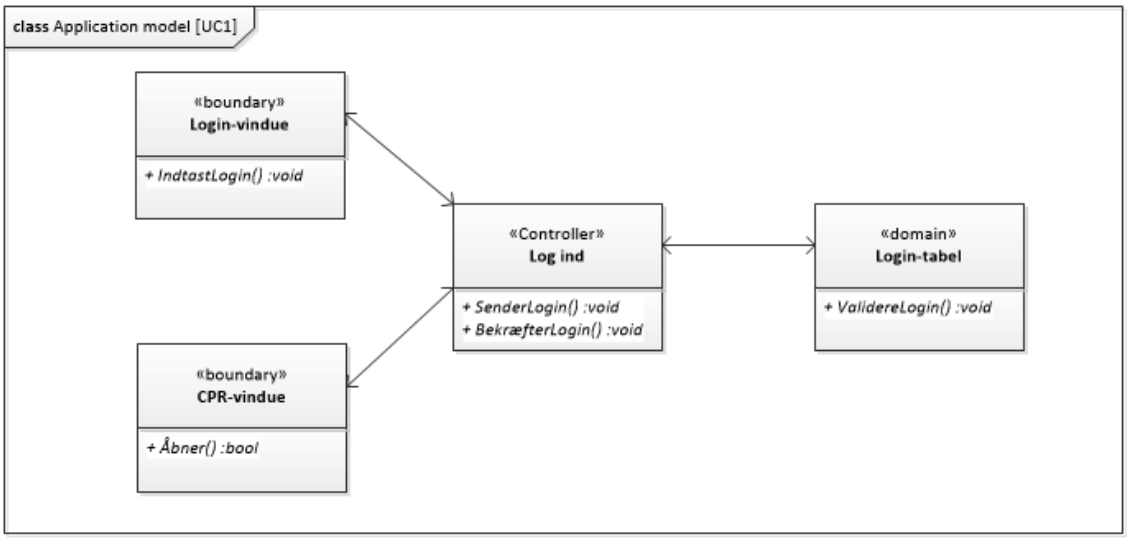
Figur 3.12: Sekvensdiagram for UC4



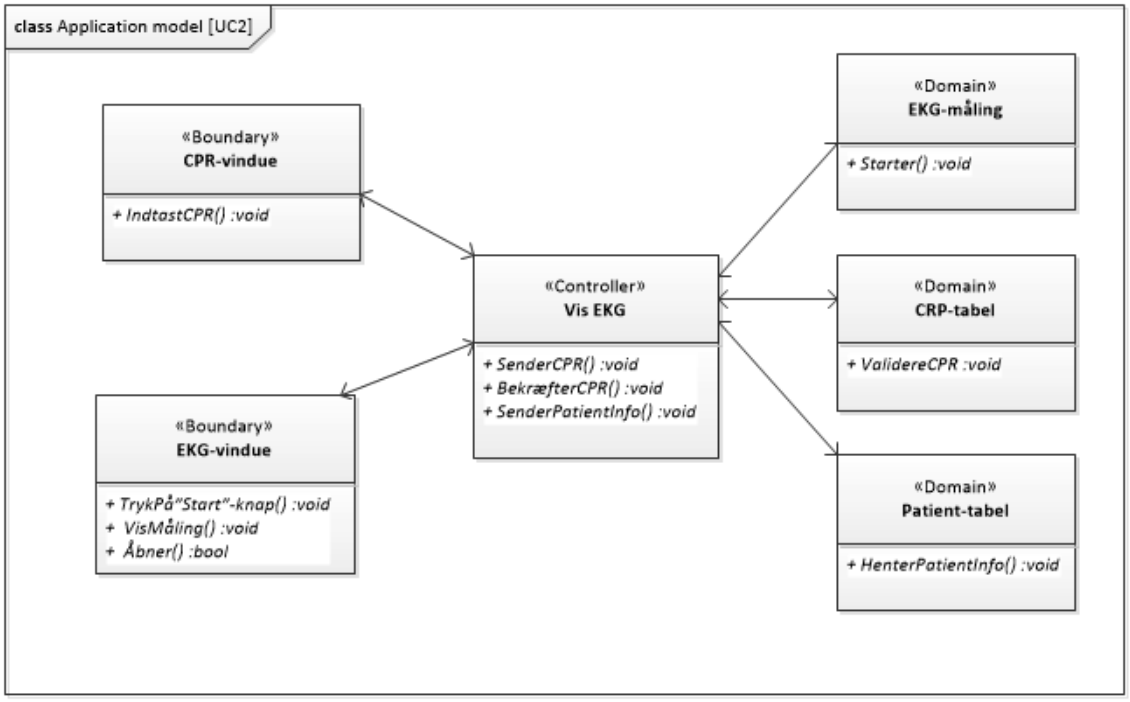
Figur 3.13: Sekvensdiagram for UC5

Opdateret Klassediagram

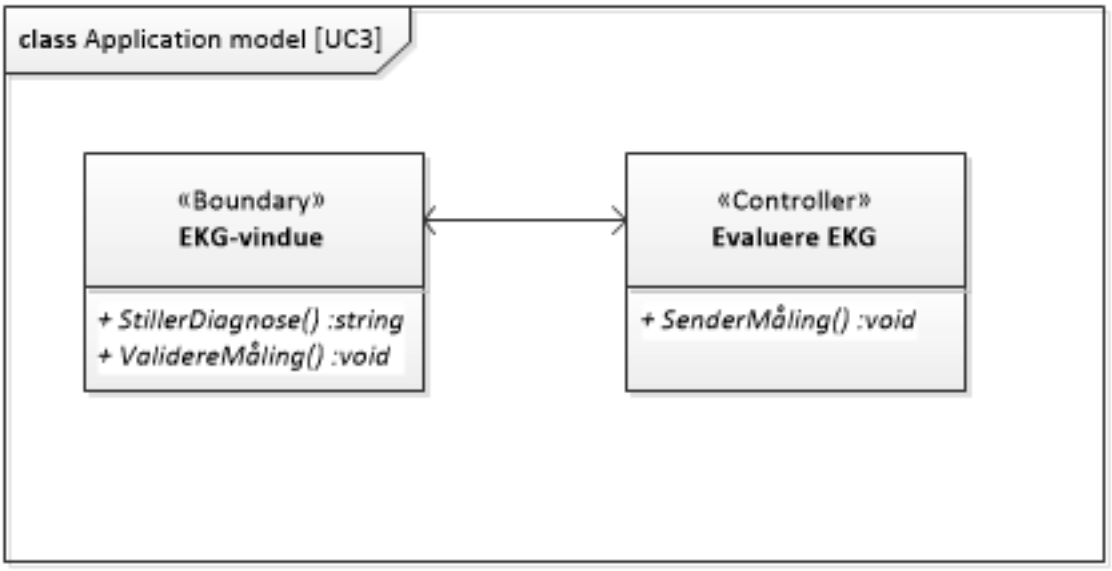
De opdateret klassediagrammer indeholder metoderne fra de dertilhørende sekvensdiagrammer - dette giver et overblik over, hvilke metoder de forskellige klasser består af.



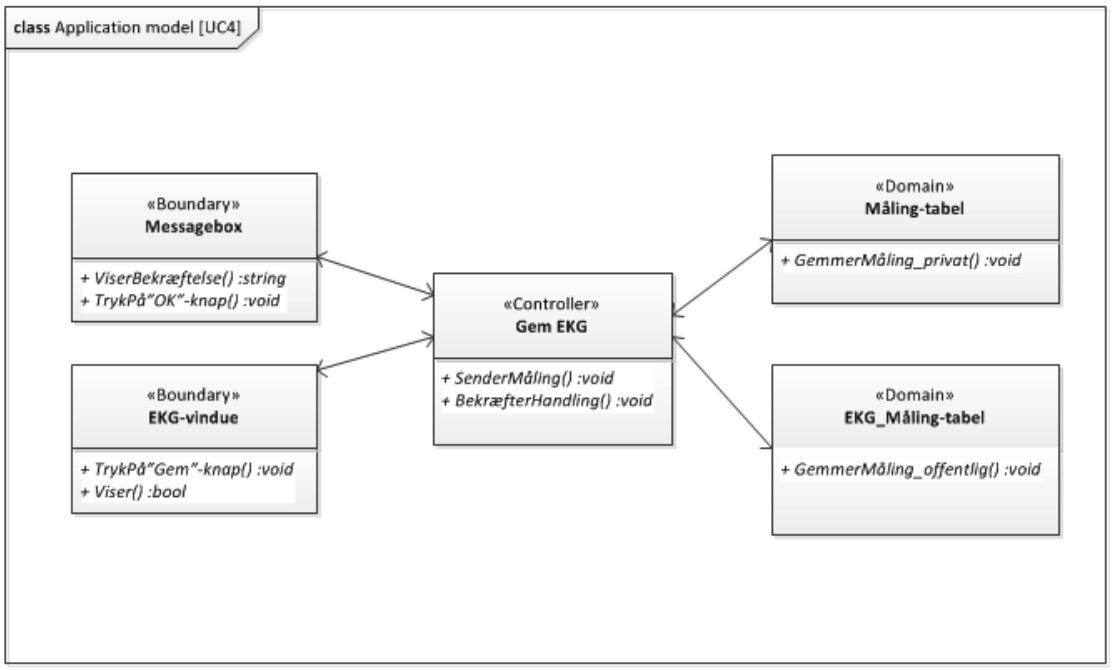
Figur 3.14: Klassediagram for UC1



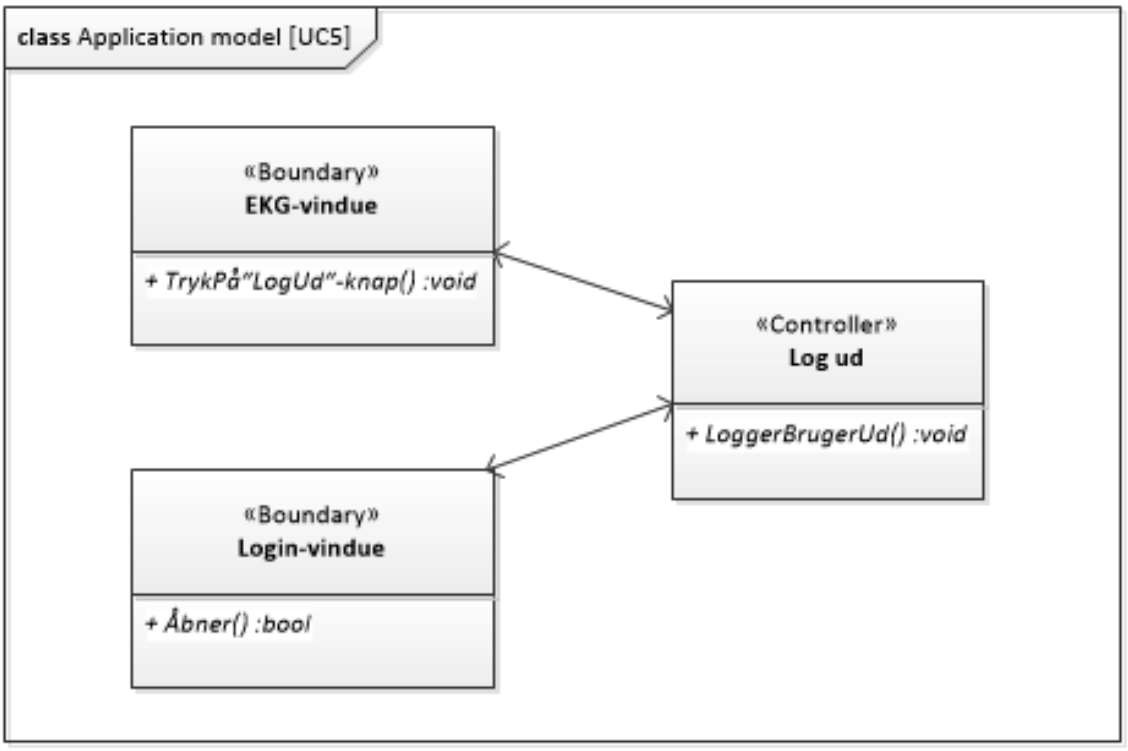
Figur 3.15: Klassediagram for UC2



Figur 3.16: Klassediagram for UC3



Figur 3.17: Klassediagram for UC4



Figur 3.18: Klassediagram for UC5

3.4 Software implementering

I dette afsnit vil ideen samt overvejelserne bag analysen af atrieflimren beskrives. Analysen fungere via et testprogram, som også her beskrives. Der vil være billeder, der understøtter og er forklarende til beskrivelsen. Kravene til programmet i forhold til at afbillede og gemme EKG-måling vil også i dette afsnit beskrives.

3.4.1 Visning af EKG-signal

Grafen bliver genereret, ved hjælp af en metode i logiklaget. Denne metode, 'Kør EKG', sætter de nødvendige forudsætninger for, at hardwaren bliver trigget til at starte en måling. Der bliver oprettet en ny sekvens, og herefter bliver navnet på enheden, samt den spænding der kommer, sat. Metoden returnerer herefter den måling, der bliver fortaget.

```
1 reference
public List<double> kørEKG()
{

    datacollector = new NI_DAQVoltage();
    datacollector.deviceName = "Dev1/ai0";
    datacollector.getVoltageSeqBlocking();
    return datacollector.currentVoltageSeq;

}
```

Figur 3.19: Kode udsnit af visning af EKG-signal

Denne metode bliver efterfølgende kaldt i grafiklaget, nærmere betegnet i Formen EKG.

```
liste = logik.kørEKG();

for (int i = 0; i < liste.Count; i++)
{
    chart1.Series["EKG"].Points.AddXY((double)i * 0.004, liste[i]);
}
```

Figur 3.20: Kode udsnit af visning af EKG-signal

Målingen bliver lagt over i en liste. Herefter køres alle punkter i målingen igennem, og bliver en for en lagt over i grafen. Disse bliver ganget med 0,004, for at konvertere samples til tid i sekunder.

3.4.2 Analyse

Denne analyse er ikke bygget op omkring den grundlæggende definition af atrieflimren, altså at atrieflimren viser sig ved små fluktuationer på baselinen. Analysen er i stedet bygget op omkring informationen at amplituden i et bestemt frekvensområde, vil være forhøjet, hvis der er mulighed for atrieflimren.

Analysen af EKG'et består grundlæggende af tre for-løkker.

Før analysen kan gå i gang, kræver det dog at den aktuelle EKG-sekvens bliver lavet om til et Furier-transformeret komplekst array. Her skal der bruges et bibliotek som indeholder metoder, som gør at der kan bruges komplekst matematik i softwaren. Dette sker ved hjælp af en metode fra biblioteket alglibnet2.

Herefter indeholder det komplekse array vektor-koordinator, som repræsenterer amplituden for alle frekvenser i signalet. I den første for-løkke, findes længden for alle vektorer en for en, og indsættes i listen amplitude.

Da det er bevist, at atrieflimren viser sig ved forhøjet signalamplitude i frekvensspekteret 300 Hz til 400 Hz, er det der, softwaren skal tjekke om amplituden ligger over tærsklen. Før selve amplitudetærsklen kan vurderes, kræves det dog at pladserne matchende frekvensspekteret, findes. Måden de er fundet, er gjort rede for, i afsnittet omkring testprogrammet.

I den næste løkke, bliver de pladser tilsvarende det valgte frekvensspektrum, lagt over i en liste for sig. Dette sker fordi analysen kun er interesseret i de amplituder som ligger i frekvensspektrummet, og det ville derfor være unødvendigt at kigge på samtlige 3600 amplituder, som er at finde i amplitude listen. Derfor ligges de udvalgte pladser i amplitudelisten over i endnu en liste, som indeholder de endelige amplitude-resultater for frekvensspektrummet i det aktuelle EKG.

Endeligt bliver der i den sidste for-løkke vurderet hvorvidt om amplituden i frekvensspekteret er højere end den valgte tærskel.

Det skal nævnes, at denne analyse ikke passer på alle signaler. Dette er der to grunde til. Den ene er, at atrieflimren kan forme et signal på mange forskellige måder. Det vil derfor være svært at lave en analyse, som vil kunne dække over alle signaler. Den anden ting er, at alle mennesker er forskellige. Vores hjerterytmer kan have forskellige baselines, og vil derfor ligge forskelligt spændingsmæssigt.

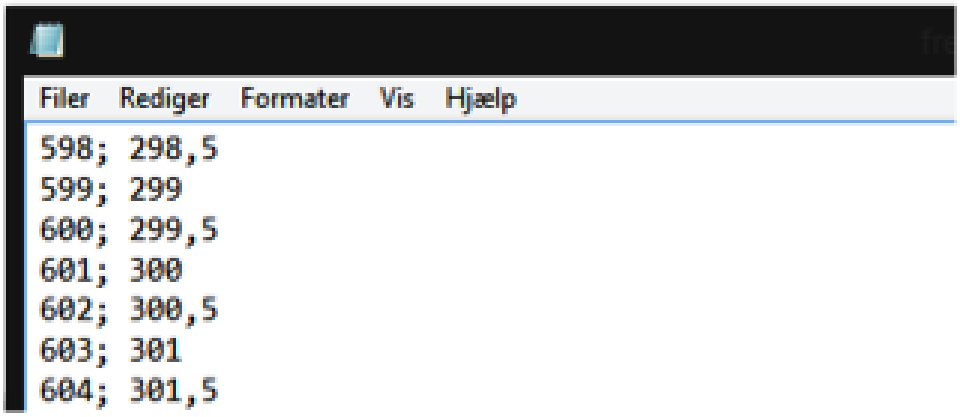
3.4.3 Testprogram

Den essentielle i analysen, er amplitudetærskelen. For at kunne vurdere tærsklen, kræver det, at man har værdier for både et rask og et sygt EKG, og ud fra dette vurderer hvornår et signal har så betydelige ændringer i dets amplitude, at der kan være tale om atrieflimren. Til dette har gruppen valgt at skrive et testprogram. Programmet er skrevet, og brugt, i flere trin. Ligesom EKG-softwaren, er testprogrammet forbundet med matematikbiblioteket og STPrj2LibNI-DAQ-biblioteket.

Fremgangsmetoden for behandling af signalet er i første omgang den samme som ved den reelle analyse; der laves en furier-transformation og derefter regnes arrayet om til en liste af amplituder. Ud over dette, laves der en liste af frekvenser. Dette sker ved dividere antal samples med antal pladser i arrayet, og derefter dividere resultatet med 2. Dette gøres for at fjerne fordoblingen af frekvenserne.

De ønskede lister bliver herefter udskrevet i en tekstfil.

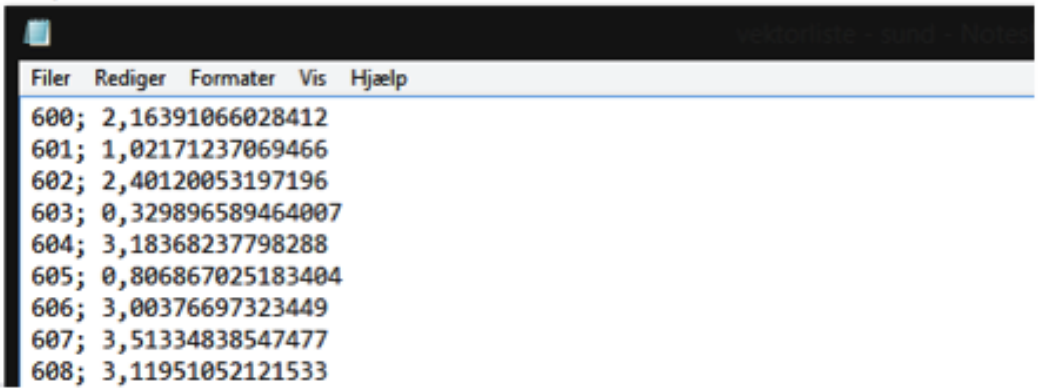
```
for (int i = 0; i < array.Length; i++)
{
    double frekvens = i * (sample / array.Length / 2.0);
    frekvensliste.Add(frekvens);
}
```



Figur 3.21: Kode udsnit af testprogram

På figur (nummer) kan der ses, hvor det valgte frekvensspekter ligger i listen. Herefter bekræftes det, at det valgte frekvensspekter, altså fra 300 Hz til 400 Hz, starter på plads 601 og slutter på plads 801. Derved skal der forstås, at amplituderne for frekvensspekteret altså også ligger på de samme pladser i deres liste. Herefter regnes længden af vektorerne ud, og disse bliver derefter også skrevet ud i en fil. Dette kan ses på figur (nummer).

```
for (int i = 0; i < array.Length; i++)
{
    double amp = (Math.Sqrt(Math.Pow(array[i].x, 2)+Math.Pow(array[i].y, 2)));
    amplitude.Add(amp);
}
```



Figur 3.22: Kode udsnit af testprogram

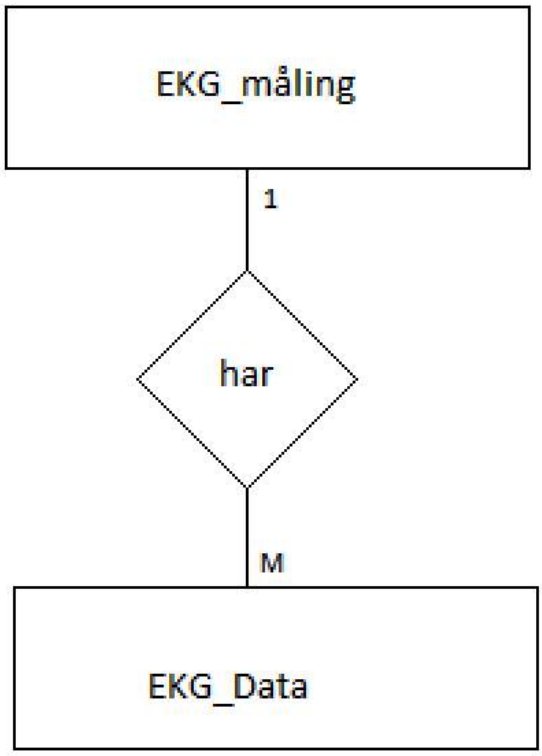
Det samme gøres for et sygt EKG. Herefter sammenlignes amplituderne, og der kan vurderes en tærskel. I vores projekt er tærsklen vurderet til at være 5.6 V.

3.4.4 Lagring i database

Patienternes data og målinger gemmes i to forskellige databaser. Den ene database er oprettet af semestergruppen og den anden er en offentlige database udleveret af vejlederen i slutningsfasen af projektet. Den offentlige database bruges til at indrapportere EKG målinger til brug fra forskning og udveksling af EKG målinger. Der er brugt SQL til at gemme og hente patienternes data og EKG målinger. Herunder vil der kort blive beskrevet hvordan QL-koden er implementeret.

Offentlig database

I den offentlige database findes der to tabeller med navnene EKG_måling og EKG_Data. Associationen mellem de to tabeller illustreres som følgende. Figur 1 viser at en patient kan have mange EKG målinger.



Figur 3.23: Association mellem EKG_måling og EKG_Data

Opsætning til den offentlige database sker ved at navnet til databasen erklæres vha. en private access specifier i datalaget. Herefter bruges en default constructor til at skabe forbindelse. Her fra er forbindelsen til den offentlige database skabt.

```
private const String db2 = "F15ST2PRJ2OffEKGDatabase";


public Datalag()
{
    conn2 = new SqlConnection("Data Source=10.29.0.29;Initial Catalog=" + db2 + ";Persist Security Info=True;User ID=" + db2 + ";Password=" + db2 + "");

}
```

Figur 3.24: Kode udsnit fra lagring i database

Metoden "GemEKGDATA"gemmer patientens data i den offentlige database. De data som en patient kan ses som parametre i metoden. Disse data bliver lagt ind i de tilhørende kolonner i tabellen EKGDATA. Der er valgt én parameter til at demonstrere indsætning af en patients data i tabellen. De resterende parameter behandles tilsvarende.

```
public bool GemEKGDATA(byte[] måling, float samplerate_hz, long interval_sek, string
data_format, string bin_eller_tekst, string maaleformat_type, DateTime start_tid)
        {

conn2.Open();

cmd = new SqlCommand("insert into EKGDATA(raa_data, samplerate_hz, interval_sec,
data_format, bin_eller_tekst, maaleformat_type, start_tid) values (@raa_data,
@samplerate_hz, @interval_sec, @data_format, @bin_eller_tekst, @maaleformat_type,
@start_tid)", conn2);

            SqlParameter param1 = new SqlParameter();
            param1.ParameterName = "@raa_data";
            param1.Value = måling;
            cmd.Parameters.Add(param1);

        }
```

Figur 3.25: Kode udsnit fra lagring af database

Tabellen EKGMAELING behandles på samme måde som tabellen EKGDATA.

Privat database

Semestergruppens database indeholder tabellerne login, måling og patient. Indsættelsen af data er foretaget manuelt dvs. uden brug af SQL-kode.

db_owner.Patient [Data] | EKG.cs | Form1

Max Rows: 1000

	Navn	Cpr	PatientID
▶	Brian	2912	2
	Sara	0101	3
	Lise	0104	4
	Malene	1208	5
	Test	123456-7890	6

Figur 3.26: Eksempel på tabel, Patient

Forbindelsen til denne databaseserver oprettes som tideligere eksempel, med den offentlige database.

```
private const String db = "F15ST2ITS2201405722";
1 reference
public Datalag()
{
    conn = new SqlConnection("Data Source=webhotel10.iha.dk;Initial Catalog=" + db + ";Persist Security Info=True;User ID=" + db + ";Password=" + db + "");

}
```

Figur 3.27: Kode udsnit af privat databse

SQL-koden i metoden getKode går ind tabellen Login og vælger en kode som tilhører et bestemt navn. Hvis koden matcher det tilhørende navn så returner den det ønsket navn og kode.

```
public int getKode(string navn) //Slår den indtastet kode op
{
    int resultat = 0;
    cmd = new SqlCommand("select Kode from Login where Navn ='" + navn + "'", conn);
    conn.Open();
    rdr = cmd.ExecuteReader();

    if (rdr.Read())
    {
        resultat = rdr.GetInt32(0);
    }

    conn.Close();
    return resultat;
}
```

Figur 3.28: Kode udsnit af privat database

De resterende SQL-kode for tabellen patient og måling i datalaget medtages ikke her, grundet deres sammenlighed med den overstående beskrivelse. Nærmere beskrivelse

henvises til xxxx.

Acceptest 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
	Hovedscenarie			
1.	Indtast username "moh04"samt password; 1234	Username- og passwordboks bliver udfyldt	Som forventet	✓
2.	Tryk på "Login"-knappen	Login bliver godkendt. Login-vinduet lukkes ned mens CPR-vinduet åbnes	Som forventet	✓
	Exentions			
2a.	Username eller password er forkert	Besked vises på skærmen med tekst, der informerer om, at brugernavn eller password er forkert	Som forventet	✓

Tabel 4.2: Accepttest af Use Case 1.

4.1.2 Use Case 2

Vis EKG

	Test	Forventet resultat	Faktiske observationer	Godkendt
	Hovedscenarie			
1.	Indtast virtuel patients CPR-nummer; 123456-7890	CPR-nummerboks bliver udfyldt	Som forventet	✓
2.	Tryk på "Ok"-knappen	CPR er gyldig. CPR-vinduet lukkes ned mens EKG-vinduet åbnes	Som forventet	✓
3.	Tryk på "Start ny måling"	Målingen startes i EKG-vinduet	Som forventet	✓
4.	EKG-data illustreres på en graf	En analyserebar graf fremvises i EKG-vinduet	Graf vises efter ca. 20 sekunder	✓
2.a	CPR-nummeret findes ikke. Besked vises med tekst, der informerer om, at CPR-nummeret ikke er gyldigt	Nyt CPR-nummer indtastes	Som forventet	✓

Tabel 4.3: Accepttest af Use Case 2.

4.1.3 Use Case 3

Evaluer EKG

	Test	Forventet resultat	Faktiske observationer	Godkendt
	Hovedscenarie			

1.	Validere programmets analyse af EKG-signalet	Det er muligt at se små fluktuationer, som kan aflæses på EKG-grafen	Grafen er analyserbar, dog er det ikke de små fluktuationer som analyseres, se fejlrapport i bilag	(✓)
2.	Stil diagnosen atrieflimmer	Atrieflimmer kan aflæses ud fra EKG-grafen	Som forventet	✓
	Exentions			
2a.	Atriefrekvensen er ikke i intervallet 220-300 pr. minut	Det er ikke muligt at diagnosticere atrieflimmer ud fra EKG-grafen	Hvis ikke atrieflimmer er diagnostiseret, vises besked om sundt EKG. Dog skyldes det ikke atriefrekvensen, se fejlrapport i bilag	(✓)

Tabel 4.4: Accepttest af Use Case 3.

4.1.4 Use Case 4

Gem EKG

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

Tabel 4.5: Accepttest af Use Case 4.

4.1.5 Use Case 5

Log ud

	Test	Forventet resultat	Faktiske observationer	Godkendt
	Hovedscenarie			
1.	Tryk på "log ud"-knappen	EKG-vinduet lukkes ned, mens login-vinduet fremkommer	Som forventet	✓
	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
Usability				
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	Programmet er startet op efter 14 sekunder	✓
Login-vinduet skal indholde 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	Som forventet	✓

EKG-vinduet skal indeholde en "start"-knap til at igangsætte målingen	"Start"-knappen er synlig i GUI, og ved tryk på knap igangsættes målingen	At målingen igangsættes	Som forventet	✓
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	Som forventet	✓
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	Som forventet	✓
Reliability				
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	Som forventet	✓
Performance				
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	Spændingen er op ad y aksen og tiden i sekunder hen ad x-aksen. Dog er intervallet ikke -1V til 1V, se fejlrapport i bilag	✗

Det skal være muligt at kunne scrolle igennem målingerne hen ad x-aksen	Der gennemføres en måling hvorefter der scrolles hen ad x-aksen	At der ved scrolling kan ses forskellige dele af EKG-signalet hen ad x-aksen		✓
Supportability				
Softwaren er opbygget af trelagsmodellen	Kig i koden efter data-lag, logik-lag og GUI-lag	At koden indeholder et data-lag, et logik-lag og et GUI-lag	Som forventet	✓

Tabel 4.7: Accepttest af Ikke-funktionelle krav

Bilag

Fejlrapport

- Dokumentation - 4.1.3 UC 3 punkt 1

Markeret som delvist godkendt, som følge af ny analyse metode, hvor det ikke længere er fluktuationerne der analyseres. Den nye analyse metode, indebærer analyse af amplituderne i forhold til frekvenserne, beskrevet nærmere under dokumentationen (NUMMER HVADDD?)

- Dokumentation - 4.1.3 UC 3 Extension punkt 2.a

Markeret som delvist godkendt, som følge af ny analyse metode, hvor det ikke længere er fluktuationerne der analyseres. Den nye analyse metode, indebærer analyse af amplituderne i forhold til frekvenserne, beskrevet nærmere under dokumentationen (NUMMER HVADDD?)

- Dokumentation - 4.2 Performance punkt 1

Intervallet op ad y aksen, er ikke fra -1V til 1V, da det med et designmæssigt perspektiv, ikke giver mening, at medtage dette interval. For bedst mulige grafiske resultat, er intervallet ændret til -0,5V til 1V.