

X-RAY EQUIPMENT & QA DATABASE

INCLUDING REPORT GENERATOR

INTRODUKSJON

Dette er en programpakke som består av flere deler. Det er en

- Oversikt over røntgen- og CT-utstyr i form av en sqlite3-database
 - o Kan laste inn informasjon fra en mappe med DICOM-filer
 - o Last inn fra og lagre til CSV-filer
- Måleark til bruk i røntgen- og angiologer, som automatisk laster resultatene over i SQL-databasen (fungerer optimalt med Unfors Xi måleapparat)
- Rapportgenerator som lager automatiske rapporter fra SQL-databasen



FIGUR 1: HOVEDVINDU FOR PROGRAMMET.

OVERSIKT OVER DATABASEN

Databasen er ordnet som en tradisjonell SQL-database. Hvert hovedfelt (maskin, QA event, Tube QA, Fluoro tube QA, Fluoro Image Quality, DAP) inneholder flere tabeller, som hver inneholder flere verdier.

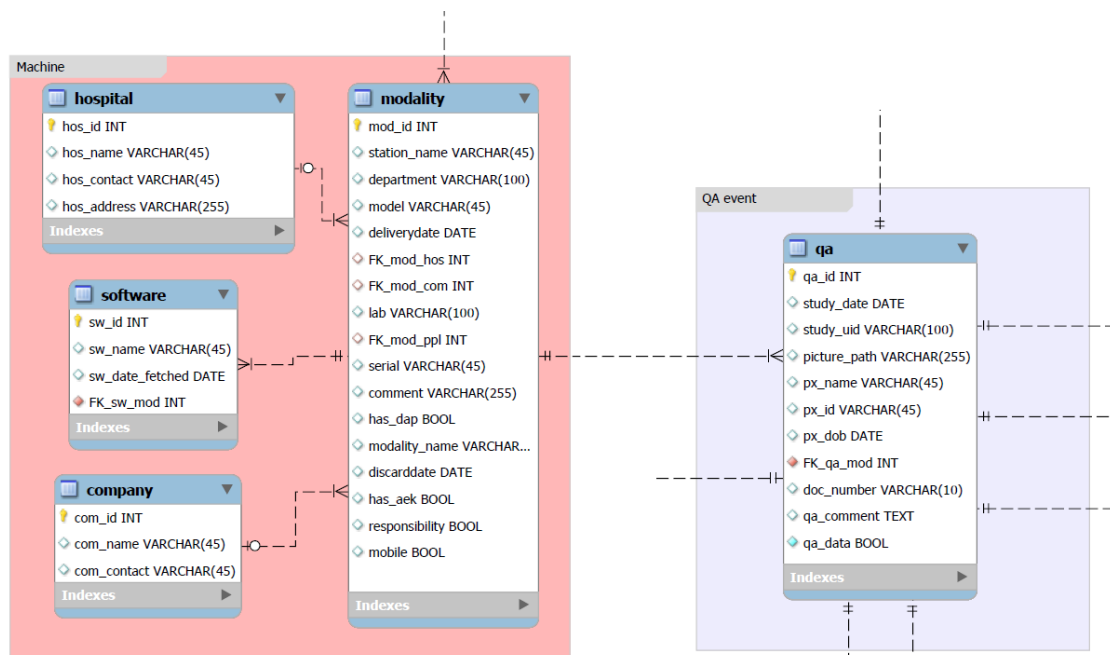
I maskin-hovedfeltet finnes tabellene «hospital», «software», «company» og «modality», se under. Her ligger informasjon om selve maskinen: Hvor den står, modell/leverandør, kontaktpersoner, når den ble plassert der med mer.

Den er koblet til QA-tabellen, som inneholder informasjon om én spesifikk måling: Hvem som var der, hvem som gjorde den, rapportkoden knyttet til målingen.

QA-tabellen er videre knyttet til en mengde andre hovedfelt, som inneholder informasjon om forskjellige målinger innen bildekvalitet, røntgenrør, DAP-meter, AEK.

Hver rekke i tabellen QA svarer til én måling, og har en ID knyttet til seg (qa_id). Denne ID-en er gjenspeilet i de forskjellige målingene gjennom foreign keys (f.eks. i DAP-tabellen er det en nøkkel FK_dap_qa, som sier hvilken QA den tilhører).

På samme måte har hver maskin (i modality-tabellen) én `mod_id`, som er gjenspeilet i alle målingene som er gjort på den maskinen (mange `qa_id`, én `FK_qa_mod`).



Se vedlagt PDF-fil for fullt kart over alle tabeller (litt utdatert), evt. bla i SQL-filen `create_sqlite_db_2014-11-24.sql`.

HELSE BERGEN
Haukeland Universitetssykehus
Avdeling for kreftbehandling og medisinsk fysikk
Seksjon for medisinsk fysikk

Statuskontroll ved Haukeland universitetssjukehus - Dagkirurgisk seksjon

Rapportnr. 14-743.

Det ble utført kvalitetskontroll på en Philips Veradisc Neo på DKS OP 9 Urologi ved Dagkirurgisk seksjon, Haukeland universitetssjukehus den 22. oktober 2014.

Kontrollen ble utført av medisinsk fysiker Daniel Aadnevik og radiograf Erik Lütherath.

Kommentar for kvalitetskontrollen: Ingen DAP

Det er registrert til kvalitetskontrollen ved dette utstyret:

Dato	Rapportnr.
2014-10-22	14-743
2013-09-02	13-573

Kontroll av røntgenrør ved fluoroskopi

Røntgenrør blir målt ved hjelp av et Unfors Xi målekammer. Avstanden mellom fokuspunkt og målekammer er 110 cm, og det blir ikke benyttet tilleggsfiltrering. Strålefeltet blendes inn til målekammeret. Slike målinger er utført der utstyret tillater manuelle innstillinger av kVp og mAs.

Nøyaktighet rørspenning (stort fokus)

Nøyaktighet i rørspenning ble målt med stort fokus og økende kV-verdier.

Krav: Bør være innenfor 5 % av nominell verdi, skal være innenfor 10 %.

Konklusjon: Godkjent.

Resultat: Avviket mellom målt og nominell kV ligger innenfor grenseverdiene for alle målinger, se tabellen under.

Set kVp	Målt kVp	kVp diff.
50	49.8	-0.4 %
60	57.2	-4.6 %
70	67.3	-3.8 %
80	77.0	-3.7 %
90	86.4	-4.0 %
100	95.4	-4.6 %
110	104.8	-4.7 %
120	113.9	-5.1 %

Reproduserbarhet rørspenning

Reproduserbarheten i kVp og dose ble funnet ved å måle rørspenning ved 80 kVp tre ganger. Deretter beregnes målingens gjennomsnittsverdi og variasjonskoeffisienter. Stort fokus benyttes.

Krav: Største avvik fra målingens gjennomsnittsverdi skal være innenfor 5 % for kVp.

Konklusjon: Godkjent.

Resultat:

Gjennomsnittlig rørspenning: 77.2 kVp

Største avvik fra gjennomsnittlig rørspenning: 0.51 %

Doser gitt under fluoroskopi

Det måles doser til pasient og detektor. Det legges 1 mm Cu plater mellom rør og målekammer for å måle dose til detektor. Ved å dekke detektor / bildeforsterker med Cu plater, for så å legge målekammer over, måler man huddose til pasient.

Dose til detektor

Krav: Dose til detektor / bildeforsterker skal ikke overstige 1 µGy/s. Målingen er utført med fokus-målekammer-avstand på 110 cm, og fokus-detektor/bildeforsterker-avstand på 115 cm. Den oppgitte måleavstanden er korrigert for avstanden, og representerer dose ved detektor.

Konklusjon: Godkjent.

Resultat: Ved alle testede programmer var dosene lavere enn grenseverdien, og høyeste verdi målt var 0.3 µGy/s. Se tabellen under.

Program	Puls/sekund	Dosemodus	Feltstørrelse	Målt dose
Extremities	8.0	fluoro	27 cm	0.06 µGy/s
Extremities	15.0	fluoro	27 cm	0.11 µGy/s
Extremities	4.0	fluoro	27 cm	0.26 µGy/s
Urology	15+	FLuoroHQ	27 cm	0.19 µGy/s
Urology	15+	Motion	27 cm	0.19 µGy/s
Thorax	15.0	Fluoro	27 cm	0.11 µGy/s
Pelvis	15.0	FLuoroHQ	27 cm	0.19 µGy/s
Spine	15.0	FLuoroHQ	27 cm	0.19 µGy/s
Head	15.0	FLuoroHQ	27 cm	0.18 µGy/s

Huddose til pasient

Krav: Huddose til pasient skal ikke overstige 1600 µGy/s ved fluoroskopi, og 2 mGy/bilde ved eksponering. Målingen er utført med fokus-målekammer-avstand på 110 cm, og fokus-detektor/bildeforsterker-avstand på 115 cm. Den oppgitte måleavstanden er korrigert for avstanden, og representerer dose ved minste fokus-hud-avstand.

Konklusjon: Godkjent.

Resultat: Ved alle testede programmer var dosene lavere enn grenseverdien, og høyeste verdi målt var 236.4 µGy/s. Se tabellen under.

Program	Puls/sekund	Dosemodus	Feltstørrelse	Målt dose
Extremities	15.0	fluoro	27 cm	140.76 µGy/s
Extremities	8.0	fluoro	27 cm	70.69 µGy/s
Extremities	4.0	fluoro	27 cm	37.04 µGy/s
Urology	15+	FLuoroHQ	27 cm	236.35 µGy/s
Urology	15+	Motion	27 cm	236.08 µGy/s
Thorax	15.0	Fluoro	27 cm	124.23 µGy/s
Pelvis	15.0	FLuoroHQ	27 cm	235.68 µGy/s
Spine	15.0	FLuoroHQ	27 cm	207.72 µGy/s
Head	15.0	FLuoroHQ	27 cm	235.95 µGy/s

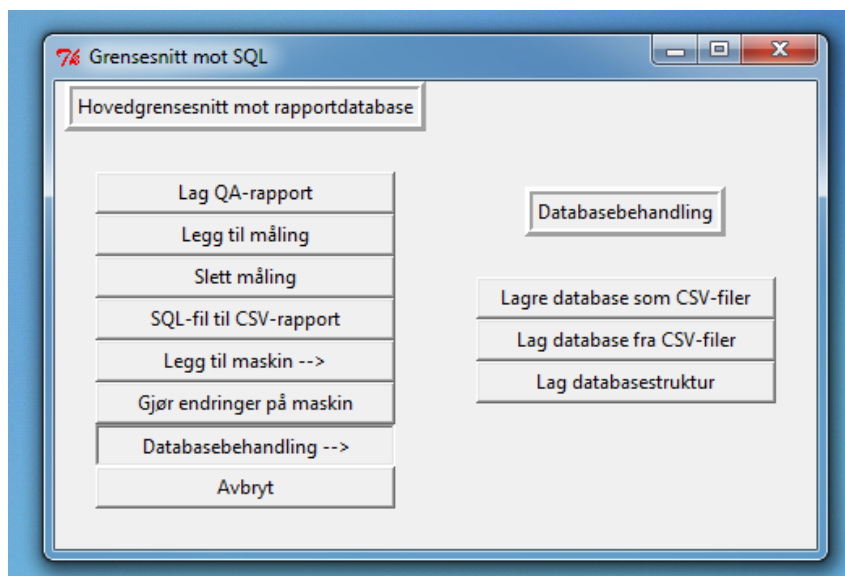
FIGUR 2: RESULTATDOKUMENT FRA MÅLING

INSTALLASJON

Siden programmet baserer seg på sqlite er det ikke nødvendig å kjøre en SQL-server: Hele databasen ligger som en fil kalt `sqlite_xray.db`. Jeg anbefaler programmet [SQLite Database Browser](#) for å bla gjennom og gjøre småendringer i databasen. Man trenger også en Python-installasjon, f.eks. fra [Python\(x,y\)](#).

Man kjører programmet med `main.py`. Man får opp vinduet som vist i Figur 1.

For å fylle opp databasen med *struktur* (tomme tabeller og regler), kjøre SQL-filen som ligger i hovedmappen. En enkel måte å gjøre det på, er å åpne `main.py`, og velge *Databasebehandling* → *Lag databasestruktur*, som vist under. Dersom denne prosessen gjentas senere må det nevnes at det er litt trøbbel med filtillatelser, så da anbefales det å slette filen `sqlite_xray.db` før den dannes på nytt.

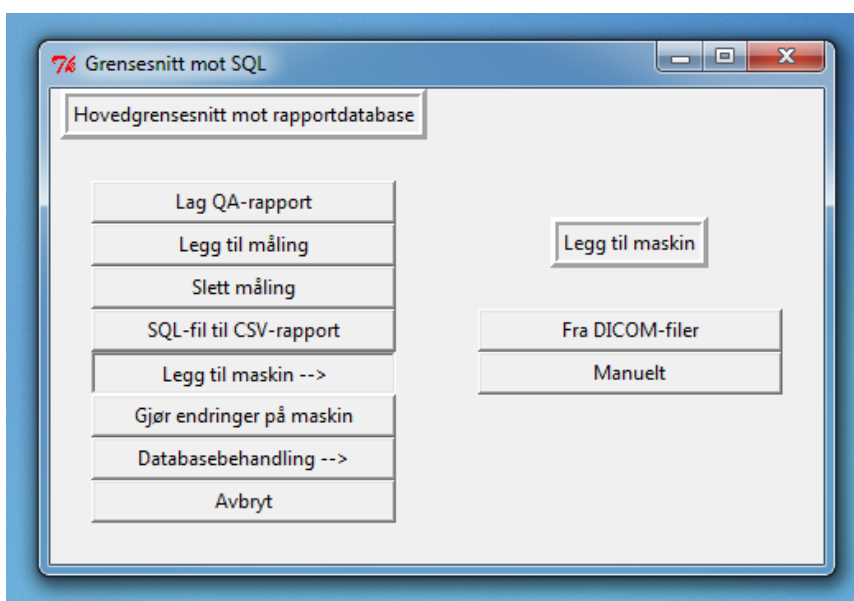


Nå gjenstår det bare å fylle databasen med **maskiner** og **målinger**.

Senere er det mulig å ta backup av databasen ved å lagre den som CSV-filer i en egen mappe. Da vil hele databasen dumpes til semikolon-separerte filer, med headere, som også kan leses i Excel. Velger man «Lag database fra CSV-filer», vil de samme filene leses inn og importeres i databasen.

Legg merke til at foreign keys vil beholdes, slik at under modalitet vil f.eks. sykehus-ID vises og ikke sykehus-navn. Bruk valget «SQL-fil til CSV-rapport» dersom man ønsker lesbare rapporter.

LEGG INN INFORMASJON OM MASKINER



Det er mulig å legge inn all informasjon om maskinene manuelt, ved å bruke *Legg til maskin* → *Manuelt*. Da får man opp et vindu hvor man blir spurt om informasjon som Stasjonsnavn (Må være unik), modell osv. Se under for komplett liste:

Legg merke til at

- Alle datoer må skrives som YYYY-MM-DD
- Velg «Legg til nytt sykehus» eller «Legg til ny leverandør» eller «Legg til ny kontaktperson» dersom du ikke finner aktuelt valg i listen. Da kommer det opp en dialog hvor man kan skrive inn navn på nytt element.
- Ansvar: Velg 1 eller 0, avhengig om det er en maskin man har ansvaret over. F.eks. dersom man ønsker å ha oversikt over alle røntgenmaskiner på en avdeling, men bare har QA-ansvar for noen av dem.
- Mobil: Om det er et mobilt apparat (stueapparat, mobil C-bue osv.)
- DAP: 1 eller 0, avhengig om apparatet har DAP-meter (la stå tom dersom det ikke er gyldig spørsmål, f.eks. CT)
- AEK: 1 eller 0, avhengig om apparatet har AEK (se over)
- Detektor: Navn på detektoren
- Modalitet: Etter eget ønske. F.eks. CT, DX (konvensjonell røntgen), XA (angiolab), RF (enklere fluorolab / mobil C-bue).
- Sykehus: Sykehusene kan ordnes etter Helseforetak: Det finnes ennå ikke et grafisk system å gjøre det på. Gå derfor inn i databasen med en db-klient (SQLite database browser), finn tabellen *helseforetak* og legg inn de aktuelle. Velg *hf_id* fra 1 og oppover, og skriv inn *hf_name*. Så kan man legge inn valgt *hf_id* på hvert sykehus i feltet *FK_hos_hf* for å fullføre koblingen.

IMPORTER FRA DICOM

Det er også mulig å forenkle prosessen ved å bruke DICOM-filer for å hente ut mye av informasjonen.

Da får man opp en fildialog hvor man kan velge DICOM-filer. Man velger alle ønskede filene, og den finner ut

1. Hvilke unike maskiner som er i filene (om det er 5 filer fra én maskin, behandler den dem kun som én maskin). Dette er basert på stasjonsnavn.
2. Alle aktuelle DICOM-merkelapper som er på hver maskin. Dette kan dreie seg om avdeling (Department), sykehus (Institution Name), modalitet (DX/CT/... fra Modality), leverandør, modellnavn, detektortype (Detector Description).
3. Siden sykehus og leverandør peker til egne tabeller, er det en fordel om maskiner fra samme sykehus/leverandør peker til de samme tabellene. Dessverre er det ikke alltid slik i DICOM-informasjonen, da både skrivefeil og forkortelser. Programmet prøver å løse dette ved å liste alle unike verdiene, og man får mulighet til å korrigere dem slik at de likner på hverandre. Se figur under:

Stasjonsnavn	Fra DICOM	Ny verdi
NFSVRTG06DR	Voss sjukehus	Voss sjukehus
MX8000	Florø Sjukehus	Florø sjukehus
CT58585	Haukeland Sykehus	Haukeland universitetssjukehus
nhds1ab1ct	Haraldsplass	Haraldsplass Diakonale Sykehus
KIHIC	Haukeland universitetssjukehus	Haukeland universitetssjukehus
HDLSLAB4CR	Haraldsplass Diakonale Sykehus	Haraldsplass Diakonale Sykehus

GJØR ENDRINGER PÅ MASKIN

Her vil man få muligheten til å endre på dataene lagret på én maskin – det er også en fin måte å slå opp informasjon om en maskin på. Man velger hierarkisk fra nedtrekkslister:

Velg maskin som skal redigeres (maksimum 1).

Velg sykehus

Haukeland universitetssjukehus

Velg avdeling

Kirurgisk Serviceklinikk

Velg maskin

DKS OP 6 OT | RF | NHBESOP2MOB02RF | Philips Veradius Neo
Hybridstuen | XA | ARTIS117003 | Siemens Artis Zeego
DKS OP 5 OT | RF | NHBESOP2MOB03RF | Philips Veradius Neo
DKS OP 4 Hjerte | RF | NHBESOP2MOB04RF | Philips Veradius Neo
DKS OP 9 Urologi | RF | NHBESOP2MOB01RF | Philips Veradius Neo

Man vil så få opp all informasjon om maskinen, hentet fra database-tabellen *modality*. Se under.

Nøkler	Gamle verdier	Nye verdier
Stasjonsnavn	ARTIS17003	ARTIS17003
Avdeling	Dagkirurgisk seksjon	Dagkirurgisk seksjon
Modell	Artis Zeego	Artis Zeego
Leveringsdato	2013-11-08	2013-11-08
Sykehus	Haukeland universitetssjukehus	Haukeland universitetssjukehus
Leverandør	Siemens	Siemens
Lab	Hybridstuen	Hybridstuen
Kontaktperson		
Kommentar		
DAP?	Ja	Ja
Modalitet	XA	XA
Kassasjonsdato?		
AEK?	Nei	Nei
Ansvar?	Ja	Ja
Mobil?	Nei	Nei
Detektor		

Her gjelder samme regler som da man laget en ny modalitet: Man velger sykehus, leverandør og kontaktperson fra en nedtrekksliste, og kan velge «Legg til sykehus/leverandør/person» for å lage en ny enhet å knytte modaliteten til.

SQL-FIL TIL CSV-RAPPORT

Under dette valget vil man få opp en fildialog, hvor man kan kjøre SQL-scripts. Det ligger allerede to stykker i mappen `sql_macros`, for å gi oversiktsrapporter over kvalitetskontroller utført i år og over maskinpark. Det er fullt mulig å lage egne SQL-filer.

Når filen kjøres, vil det lages en csv-fil i mappen `sql_macros`, som åpnes automatisk i standardprogrammet for csv (Excel).

LEGG TIL MÅLING

Velger man «Legg til måling», får man igjen opp en fildialog. Her kan man velge én eller flere filer. Dersom arket beskrevet under er korrekt brukt, vil programmet finne alle aktuelle måleparametere, -data og informasjon om hvilken lab det er målt på. Den aktuelle laben må eksistere i databasen.

SLETT MÅLING

Ved å slette en måling, vil både *qa*-tabellen samt alle tilhørende tester (som har *qa_id* som foreign key) slettes gjennom en kaskade-regel i SQLite. Man velger rapport på samme måte som når man velger en maskin å gjøre endringer på, bare at det er ett ekstra nivå i valget (rapportnivå).

LAG QA-RAPPORT

I dette valget ligger mye av styrken til database-systemet. Ved hjelp av [ReportLabs](#) open source-løsning, vil Python lage en PDF-rapport av måledataene som ligger på valgt kvalitetskontroll (Fordelen framfor å bruke f.eks. LATEX er at man slipper å ha installert en suite – hele koden til ReportLab er inkludert i pythonkoden som er følger med i dette programmet).

Etter å ha valgt rapport på samme måte som man sletter måling (hierarkisk sykehus -> avdeling -> maskin -> rapport), vil systemet hente opp alle data, og putte dem inn i en PDF-fil, som vil åpnes etterpå. Det er noen muligheter for å velge hvordan innholdet presenteres: «Kort rapport» fjerner all beskrivelsestekst og korter ned rapporten til én side, og «Small Leeds table» brukes dersom det er gjort mange

bildekvalitetsmålinger, og tabellen fyller mer enn sidebredden – med dette valget vil skriftstørrelsen bli mindre og tabellen krympes.

Dersom det er lagt inn personer i Excel-arket som ikke finnes i databasen, vil de legges til. I Python-promptet vil det komme et spørsmål om jobben til personen, og det er teksten som vil stå foran navnet i rapporten: F.eks. «radiograf» eller «medisinsk fysiker». Tittelen lagres også i databasen, så man slipper å skrive det inn hver gang.

Selve innholdet i filen kan endres på, men her er det nok mye rom for forbedringer: Alle endringer må gjøres i python, men ved å se på eksisterende kode bør det være selvforklarende. Se forklaring til `sql2pdf.py` under.

MÅLINGSARK TIL BRUK I LABEN

Målingsdata plasseres en Excel-file basert på `Målingsark.xlsm`, som er vedlagt dette programmet. Det arket inneholder vanlige tester utført ved kvalitetskontroll av røntgen- og angioutstyr – om noe nedkrympet. Man kan kopiere data direkte fra unfors Xi, det plasseres da i arket *Rådata-røntgenrør* for konvensjonell røntgen og *Rådata-gjennomlysning* for angio- og andre C-buer.

Info-fanen

Arket matches mot databasen ved hjelp av stasjonsnavnet, eller alternativt Meridanr. (eller liknende intern seriekode) dersom stasjonsnavn ikke finnes. Det anbefales å alltid bruke et unikt stasjonsnavn på utstyret, som kan finnes i DICOM-tag (0008, 1010). Leverandør/modell er ikke nødvendig å skrive inn, men det er hjelpsomt om man ikke er sikker på utstyret.

Dato skrives inn som «YYYY-MM-DD», og dagens dato i korrekt format hentes automatisk med knappen *Oppdater*.

Fysikere sier hvem som er med under målingen. Fullt navn kan brukes, eventuelt intern forkortelse. Da vil det gjøres en databasespørring mot *people*-tabellen, hvor informasjon om yrke og fullt navn hentes ut – det brukes i PDF-rapporten.

Skriv inn en kommentar her om det er noe som gjelder for hele kvalitetskontrollen. Det vil være enkelt å bla gjennom kommentarer i databasen, samt at den vil vises i PDF-rapporten. F.eks.: «Måleutstyr for DAP defekt, måling utføres på nytt i mai».

Rapportnr. er det interne rapportnr. for kontrollen. I Bergen bruker vi YY-###, hvor ### er løpenummeret. Dersom det ikke fylles ut, vil det automatisk økes til YY-###+1 (så den første vil være YY-001).

Legg merke at det er mulig å legge til kvalitetskontroller man ikke har data i riktig format fra (f.eks. CT om man vil ha oversikt over når hva er gjort). Det gjør man rett i SQL-databasen fra en ønsket klient: Legg til ønsket *study_date* i formatet YYYY-MM-DD og koble det opp mot riktig modalitet gjennom *FK_qa_mod*. Der er det også mulig å legge inn kommentar. For riktig bokføring kan man også sette *qa_data* = 0 for å vise at det ikke er koblet opp noen tester mot kvalitetskontrollen.

Bildekvalitet

Her velges ønsket Leeds-plate (PTWLUT, TOR CDR eller TOR 18FG). Det skrives inn brukt program, dosemodus og feltstørrelse, og antall sirkler og linjepargrupper skrives inn slik det er målt: På kontrollrom / lab, og live / last image hold. Vi bruker også denne bildekvalitetsanalysen for konvensjonell røntgen, hvor vi

tillater oss å zoome helt inn i etterkant. Ut i fra hvilken modalitet apparatet er «DX/DR/CR» eller «RF/XA» vil kravene endres (strengere for DX/DR/CR). Ikke alle feltene fylles ut, og det er program/dosemodus/feltstørrelse/kontrast+lpmm i hvite felter SAMT kommentarfeltet som følger med videre i databasen.

Røntgenrør

Skriv inn Fokus-Kammer-Avstand (FKA) i cm, samt filter som er brukt (0 mm Cu). Om noen kommentarer for testen, skriv det inn i det blå feltet.

Så er det en fordel om man kobler på Unfors / barracuda. For grovfokus plasserer man Unfors-markøren i felt A1 i «Rådata-røntgenrør», og for finfokus er det A26. For Barracuda får dere prøve dere fram (også i forhold til hvert datafelt skal ligge). Så lenge måleverdiene kommer riktig inn på arket blir de kopiert korrekt videre til SQL, med fokus på kV_p (satt), mA (satt), ms (satt), kV_p (målt), dose, ms og HVL.

Gjennomlysning

Se over. Til venstre på arket gjøres en forenklet rørtest, hvor Unfors-data plasseres i A1 for grovfokus og A14 for finfokus (i arket «Rådata-gjennomlysning»).

For dose til detektor er det viktig å skrive inn Fokus-Kammer-Avstand og Fokus-Detektor-Avstand, samt filtrering (vi bruker 1 mm Cu, som vi legger på røret, slik at vi har rør – Cu – Unfors – Bildedetektor). Rasterfaktor kan også skrives inn her, som korrigeres for. Bruker man Unfors kan data plasseres «Rådata-gjennomlysning», A19.

For dose til pasient (hud) plasseres 1 mm Cu mellom Unfors og bildedetektor, gjerne slik at det dekker hele platen (og tvinger opp automatikken). Nå bør man i tillegg også skrive minste Fokus-Hud-Avstand, 30 cm hvor laterale projeksjoner er mulige eller hvor det ikke er bord. Unfors-data går inn i A32, samme ark som i sted.

AEK

Her er det mulig å utføre AEK-målingen på flere måter. Enten kan man måle dose vha. Unfors underveis, og skrive dosen rett inn i feltet «Dose». Alternativt kan man lage en tabell over dose ved gitt mAs (eller ms ved fast mA), hvor man først måler med faste verdier blendet helt inn til Unfors, og lager en konverteringstabell (1 mAs = 2 µGy, 4 mAs = 8 µGy osv...). Så blander man helt ut, tar bort Unfors og skriver inn mAs-verdi for hver eksponering (ikke satt, men oppgitt i bildet etter eksponering). Dosen vil automatisk regnes ut ved hjelp av konverteringstabellen og føres inn i «dose»-feltet. Husk FDA og skriv kommentarer om det blir nødvendig.

DAP

Velg kV-verdier man allerede har målinger på fra Røntgenrør / Gjennomlysning. Da vil HVL komme med automatisk, og verdiene vil sees opp mot både kV_p og HVL.

Skriv inn DAP-verdier man får oppgitt fra eget DAP-meter i «DAP uten bordattenuasjon». De vil korrigeres mtp. arket «DAP kalibreringsbevis», og beregnede «ekte» verdier vil dukke opp i «Beregninger mht.». Så skrives DAP-avlesningene fra modaliteten inn i «Apparat-avlesninger», og resultatet vil dukke opp i bunnen (og være klart for kopiering til SQL). En avlesningskalkulator er tilgjengelig om apparatet ikke nullstiller mellom hver eksponering.

Spredt Stråling

Se informasjon i arket. Vi bruker Unfors Xi med Dose Survey for å gjøre slike målinger, men da måleutstyret er såpass nytt har vi ennå ikke funnet veldig gode rutiner for det.

DAP-kalibreringsbevis (og –data)

Kopier inn data fra DAP-kalibreringsbevis her.

SQL-arket

Dette arket fyllet automatisk med målverdier basert på posisjon i de forskjellige arkene: Så ikke gjør så mye med dem (legg til rader, flytt, osv.).

TEKNISK INFORMASJON: PYTHON

For å kjøre programpakken trengs følgende Python-funksjoner:

- `Sqlite3`
- `Tkinter`
- `Dicom`
- `Numpy`
- `Subprocess`
- `Webbrowser`
- `Openpyxl`

Det anbefales å bruke Python(x,y) for å installere Python, og her kan man huke av funksjonene over (dersom de ikke er påtvungne).

FORKLARING TIL PROGRAMFILENE

Forklaring til filen `main.py`:

- 1) Programmet kjøres med denne filen.
- 2) Dersom det ikke finnes en backup-fil av `.db`-filen denne uken (basert på ukenr.), lages det en ny. Den kopieres til `backup\dump\sqlite_xray_backup_YYYY-wWWW.db`.
- 3) Det vil raskt lages en et objekt basert på klassen `MainMenu`, som er en `Tkinter`-meny definert i `classes.py`. Denne menyen vil returnere en tekststreng kalt `state`, som beskriver hva som skal skje videre. Mulige tilstander er: `CreateReport`, `addQA`, `deleteQA`, `runSQL`, `addModalityDICOM`, `addModalityManual`, `editModality`, `csv2sql`, `sql2csv`, `createSQL` og `cancel`.
- 4) Basert på tilstanden, vil enten kommandoen kjøres i `main`, eller et nytt program vil kjøres. Ofte vil det først genereres en ny GUI fra `classes.py` som spør etter mer info: Finn maskin eller rapport, hvilke parametere, finn fil osv.

Forklaring til filen `sql2pdf.py`:

- 1) Etter å ha hentet inn aktuelle biblioteker og definert noen variabler (dokumentforfatter, logo som skal brukes), finner den kvalitetskontrollen basert på rapportnr.
- 2) Så vil den hente inn grunnleggende informasjon om kvalitetskontrollen fra tabellen `qa`: Dette er info som dato, hvem som var med, `qa_id` for å finne tilhørende tester osv.

- 3) Den vil gå gjennom hver eneste mulige test og lage et flagg som sier om testen var gjort eller ikke. Dette er en hjelp for senere, hvor enhver test som ikke er gjort hoppes over, og det skrives ikke overskrift, tekst eller tabeller for den.
- 4) For hver test, dersom testen er gjort: Skriv header, lag introduksjonstekst. Hent inn alle data for testen, og se om hvorvidt resultatet er under kravene. Skriv resultattekst basert på resultatet. Dersom det er aktuelt, vis tabell for testen.
- 5) Lagre PDF-fil, og åpne den i standardprogram for PDF (ved hjelp av Pythonmodulen *webbrowser*).

Forklaring til filen `classes.py`:

- 1) Her ligger alt det grafiske, som baserer seg på TKinter. I tillegg ligger klassen `Modality` her, som ble laget tidlig i prosessen for å ta hånd om informasjon fra DICOM før det ble sendt til databasen.
- 2) Klassen `MainMenu` inneholder hele introduksjonsmenyen. Med litt erfaring fra TKinter bør det være rett frem å forstå hva som skjer: Man må lage Containers som hører til hovedframe, og plassere alle objekter i sin Container for at de skal plasseres riktig sted i skjermbildet.

Objekter må først defineres, og så må de pakkes (plasseres). Her finnes også rutiner for å skape og kollapse en ekstra meny til høyre: Det gjelder for funksjonen *addModalityChoice* for å velge mellom manuell eller DICOM-basert modalitetsinput, samt for *editSQL*, hvor man kan velge mellom å lage SQL fra SQL-fil, ta backup til CSV eller hente inn backup fra CSV.

Dersom det ikke eksisterer noen modaliteter i databasen, vil valg som baserer seg på at det finnes modaliteter være grået ut.

- 3) Klassen `ChooseReports` inneholder valgmenyen for å velge modalitet og rapport for å legge til, redigere, slette eller lage rapport. Den bruker dynamisk oppdatering av rullegardinsmenyer for å begrense valgmulighetene hierarkisk. Dette skjer i funksjonene *updateoptions_hospital*, *updateoptions_department*, *updateoptions_modality* og *updateoptions_report*. Rullegardinmenyene kobles til de funksjonene ved hjelp av å sette en *trace* på variablene:

```
self.variable_hospital = StringVar(self)
# self.dict kommer fra en organisering av data fra SQL til et dictionary
self.variable_hospital.trace('w', self.updateoptions_hospital)
self.optionmenu_hospital = OptionMenu(self.boxframe, self.variable_hospital, *self.dict.keys())
```

- 4) Resten av klassene følger same standard og bør være selvdokumenterende.

Forklaring til filen `dicom2sql.py`:

- 1) Får filnavn fra `main`, og åpner en databasekobling
- 2) Leser all aktuell info fra DICOM-filen: Stasjonsnavn, leverandør, dato, software, institusjon (sykehus), modalitet, modellnavn, avdeling, detektortype
- 3) Legger infoen inn i en `Modality`-klasse definert i `classes.py`. Dette gjøres i en løkke dersom flere filer er valgt, og det kontrolleres at ikke apparatet allerede finnes i listen over modaliteter.
- 4) Så sørger den for at alle verdier som har egne tabeller (sykehus, leverandør, kontaktperson) eksisterer i sin tabell: Om ikke, lages den.
- 5) Med alle koblingsnøkler (foreign keys) til overnevnte verdier, legges all info om maskinen inn i databasen. Dette gjøres ved linjen:

```
# insert modality into database
cur.execute("INSERT INTO modality (station_name, department, model, deliverydate, discarddate, \
lab, serial, FK_mod_hos, FK_mod_com, FK_mod_ppl, modality_name, \
has_dap, comment, has_aek, responsibility, mobile, detector) \
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?, ?)", \
```

```
(modality.getStationName(), modality.getDepartment(), modality.getModel(),  
modality.getDeliveryDate(), modality.getDiscardDate(), modality.getLab(), modality.getSerial(),  
FK_mod_hos, FK_mod_com, FK_mod_ppl, modality.getModality(), modality.getHasDap(), modality.getComment(),  
modality.getHasAek(), modality.getResponsibility(), modality.getMobile(), modality.getDetector()))
```

- 6) Så undersøkes det om det finnes tidligere QA-er for maskinen (hovedsakelig en legacy-funksjon fra versjonen hvor tidligere apparater ble hardkodet inn for å bevare info mellom database-rens), og legger den inn om den finnes.
- 7) Commit (lagre alle transaksjoner i databasen).

Kommentarer til filen: Jeg har hatt *mye* trøbbel med unicode her. Jeg anbefaler å kjøre `main.py` fra en IDE som Spyder når man skal kjøre bruke `dicom2sql`. Da funker det, i motsetning til IDLE / cmd. Det har ikke vært samme trøbbel med de andre funksjonene, kanskje pga. interface mot `pydicom`.

Forklaring til filen `sql2csv.py`:

- 1) Denne filen har tre funksjoner: `csv2sql()`, `sql2csv()` og `createSQL()`.
- 2) `csv2sql()` åpner alle tabellene definert i listen `tables`, samt alle filene gitt ved «`csv\{tables[i]}.csv`». Headeren leses inn (første linje i CSV-fil). Hver påfølgende linje kobles mot sin header og leses inn i aktuell tabell som `for line_value in all_lines: header[i] = line_value[i]`.
- 3) `sql2csv()` leser inn alle verdiene i hver tabell, ved hjelp av «`SELECT * FROM table_name`». Bruker `cursor.description` for å få header-navn, som så skrives ut til hver sin csv-fil i samme struktur som i forrige funksjon.
- 4) `createSQL()` vil rense hele databasen og laste skjemaet på nytt fra SQL-filen i den medfølgende mappen. Det er først litt sjekker på om det finnes en database, samt popper det opp et spørsmål på om man ønsker å slette den som ligger der. Det har vært litt trøbbel med denne funksjonen, da det ikke virker som om den klarer å lukke SQL-filen etter å ha testet om den er der... Så om man vil starte databasen på nytt anbefales det å slette `.db`-filen.

Forklaring til filen `sql_connection.py`:

- 1) `exec_sql_file`: Kjører SQL-filen `sql_file`. Dette er en fiks av funksjonen under, som jeg tror gjør ting i gal rekkefølge (fra stackoverflow).
- 2) `isSQLite3(filename)`: Sjekker om `filename` er en gyldig sqlite3-fil, basert på de 16 første bytes.
- 3) Så laser den inn database-filen `sqlite_xray.db`, slik at den er tilgjengelig når man importerer `sql_connection`.

Forklaring til filen `xls2sql.py`:

- 1) Importer `openpyxl`
- 2) Åpne excel-filen gitt i en loop, på arket «SQL»
- 3) Hent inn preliminære data (hvem som er med på QA, modalitet, ID-er osv.)
- 4) Nå er data i SQL-fanen i Excel-arket organisert litt annerledes enn hvordan det går inn i databasen. Husk at i Excel-arket er alle måleresultatene kopiert på rams etter hvordan de er målt, og det er ikke den beste måten å gjøre det på i databasen.
- 5) Last inn alle data og konverter dem til SQL-formatet
- 6) Last inn i SQL (for hver tabell).
- 7) Commit

TAKK TIL:

UNN for måleark som vi har jobbet videre med!