

[UiB](#) | [Dept. of Information Science and Media Studies](#)
[UiTø](#) | [Dept. of Computer Science](#)
[NTNU](#) | [Dept. of Computer and Information Science](#)
[Telenor](#) | [Telenor R&I](#)



CAIM
CONTEXT-AWARE IMAGE MANAGEMENT

BergenBy Database & Metadata Editor

Marius Langøy

Aug.18, 2008

CAIM-UiB

TR-5

Innhold

1. Prosjektmål:	3
1.1 Hvilken funksjonalitet som skal utvikles:	3
1.2 Databasestruktur:.....	3
1.3 Metadata Editor:.....	4
2. Design: Fremgangsmåte, verktøy	5
2.1 Fremgangsmåte:	5
2.2 Beskrivelse av databasen:	6
3. DDL for databasen:.....	8
4. Innhenting av bilder:	8
4.1 Bildeoversikt:.....	9
5. Verktøy:	10
6. Metadata Editor Prototype	11
6.1 Videreutvikling av Maritim Editor?.....	11
6.2 ImageEdit:.....	11
6.2.1 Funksjonalitet:.....	12
6.2.2 Utvikling:	13
6.2.3 Kjente bugs:	16
7. Future development/tasks	18
8. User manual, with screen shots.....	19
8.1 MetadataEditors:	19
8.1.1 TempEdit:	19
8.1.2 ImageEdit:.....	24
References.....	25
Appendix:.....	25

1. Prosjektmål:

1.1 Hvilken funksjonalitet som skal utvikles:

BergenBy databasen på Bulmeurt inneholdt fra før av 70 Bilder som var knyttet til 8 Objekter. Dette var bilder som var hentet inn av dem som arbeidet med CAIM prosjektet i 2007, samt historiske bilder fra UB sin samling. Det var lagret metadata om disse i form av tittel og fotograf av Bilder, og navn, byggeår, materialer og klassifisering av Objekter. Strukturen for å legge til ytterligere metadata til Bilder og Objekter var på plass, men disse var satt til null-verdier.

Min oppgave var å utvide databasen ved å hente inn ytterligere 100-200 bilder av 10 objekter. Hvert objekt skulle ha minst 2 historiske bilder fra UB sin samling knyttet til seg, og det skulle være minst 1 tekstdokument med beskrivelse av objektene.

1.2 Databasestruktur:

Selve databasestrukturen skulle også utvides til å inneholde metadata om:

- Location (GPS of camera, GPS of central object)
- Time (date, time of photo, time of year, day)
- Weather conditions in photo
- Number of primary objects in photo
- Name(s) of central objects in photo
- Type(s) of central objects
- Activity depicted
- Keywords
- Descriptive text for central objects
- Samt utvikle nødvendig DBMS funksjonalitet for indeksering og bildesøk basert på disse metadata.

1.3 Metadata Editor:

Det var også krav om en metadata editor for å kunne modifisere metadata i bildene. Det ble foreslått å bygge videre på metadata editoren for Maritim databasen utviklet av Øistein Myge i 2007.

2. Design: Fremgangsmåte, verktøy

2.1 Fremgangsmåte:

Jeg valgte å begynne prosjektet med å utvide BergenBY databasen med ny metadata struktur. Det var derfor nødvendig med flere kopier av databasen, slik at det ikke skulle være nødvendig å jobbe mot live-versjonen som lå ute. Etter å ha spurt litt rundt fikk tilgjengelig DDL og insert setninger til alt innholdet på BergenBy, slik at vi kunne gjennomføre dupliseringer av databasen snarere enn kopieringer.

Den endelige strukturen baserer seg på følgende typer i BergenBy databasen:

IMAGE_TYPE		AUTHOR_TYPE	
Id	NUMBER	Id	NUMBER
Caption	VARCHAR2(255)	Name	VARCHAR2(255)
Photographer	REF AUTHOR_TYPE	Affiliation	VARCHAR2(255)
Texts	NT_TEXT_REF		
IMAGE	ORDSYS.ORDIMAGE		
IMAGESIGNATURE	ORDSYS.ORDIMAGESIGNATURE	CONTAINS_TYPE	
Coordinates	MDSYS.SDO_GEOMETRY	Object	REF OBJECT_TYPE
ImgDate	TIMESTAMP	Image	REF IMAGE_TYPE
Weather	VARCHAR2(50)	Location	CHAR(2)
Activity	VARCHAR2(255)		
Keywords	NT_KEYWORDS_REF		
		KEYWORDS_TYPE	
		Id	NUMBER
IMAGE_TEMP_TYPE		Word	VARCHAR2(255)
Id	NUMBER		
Caption	VARCHAR2(255)	TEXT_TYPE	
IMAGE	ORDSYS.ORDIMAGE	Id	NUMBER
IMAGESIGNATURE	ORDSYS.ORDIMAGESIGNATURE	Summary	VARCHAR2(255)
Coordinates	MDSYS.SDO_GEOMETRY	Language	VARCHAR2(50)
ImgDate	TIMESTAMP	Text	CLOB
Weather	VARCHAR2(50)	Authors	NT_AUTHOR_REF
Activity	VARCHAR2(255)		
OBJECT_TYPE			
Id	NUMBER		
Name	VARCHAR2(255)		
Age	NUMBER		
Material	VARCHAR2(255)		
Coordinates	MDSYS.SDO_GEOMETRY		
Area	VARCHAR2(255)		
Classification	VARCHAR2(255)		
Texts	NT_TEXT_REF		
Keywords	NT_KEYWORDS_REF		

2.2 Beskrivelse av databasen:

Image_Type: Selve bildene i databasen.

Id = Unik identifikator for hvert bilde.

Caption = Tittelen en velger å sette til bildet.

Photographer = Referanse til den som har tatt bildet.

Texts = Nøstet tabell med tekster knyttet til selve bildet.

Coordinates = GPS koordinatene der bildet ble fotografert.

ImgDate = Dato bildet ble fotografert.

Weather = Værforholdene i bildet.

Activity = Aktiviteten som foregår i bildet.

Keywords = Nøstet tabell med nøkkelord knyttet til bildet.

Image_Temp_Type: Midlertidig bilde. Når et bilde blir gitt som søkebilde, eller når et bilde blir fotografert av Geocam, blir de overført til denne tabellen.

Author_Type: Fotografene av bilder, og forfatterne av tekster.

Id = Unik identifikator for hver forfatter.

Name = Forfatterens navn.

Affiliation = Hvilken organisasjon forfatteren hører til (f.eks UiB).

Contains_Type: Knytter sammen et bilde med et objekt.

Object = Referanse til et objekt.

Image = Referanse til et bilde.

Location = Hvor på bildet et objekt befinner seg.

Keywords_Type: Nøkkelord

Id = Unik identifikator for nøkkelord.

Word = Nøkkelordet.

Text_Type: Tekstene i databasen.

Id = Unik identifikator for hver tekst.

Summary = Oppsummering av teksten.

Language = Språk teksten er skrevet på.

Text = Selve teksten.

Authors = Nøstet tabell med referanse til forfattere av teksten.

Object_Type: Objektene i databasen.

Id = Unik identifikator for hvert objekt.

Name = Navnet på objektet.

Age = Året objektet ble konstruert.

Material = Materialene objektet er laget av.

Coordinates = GPS koordinatene der objektet befinner seg.

Area = Området objektet befinner seg i.

Classification = Hvilken type objektet er.

Texts = Nøstet tabell med referanser til tekster om objektet.

Keywords = Nøstet tabell med referanser til nøkkelord knyttet til objektet.

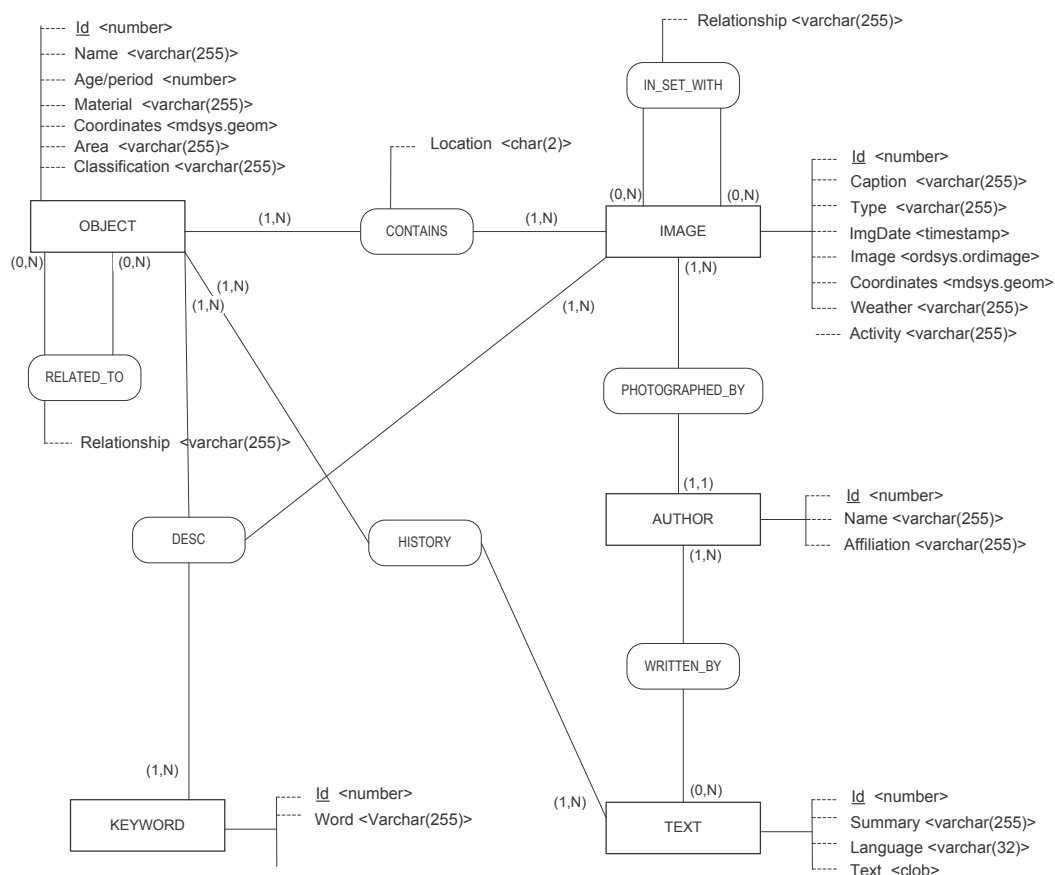
I selve strukturen ligger i tillegg typene "Related_To" og "In_Set_With" som ligger igjen fra den tidligere versjonen av BergenBy. Det finnes også en prosedyre RELATE_IMAGES, som kan gjøre dette. Disse typene benyttes foreløpig ikke til noe i databasen, men kan brukes dersom en ønsker å knytte objekter eller bilder til hverandre i framtiden. Tabellene CBIRGPS_TEMP og CBIRTBIR_TEMP brukes av VISI prototypen til beregning av scores.

Viktige prosedyrer:

HTTP_IMPORT_IMAGE: Prosedyre som legger inn et bilde fra en http kilde inn i Image tabellen..

HTTP_IMPORT_TEMP_IMAGE: Prosedyre som legger inn et bilde fra en http kilde inn i Image_Temp tabellen.

3. DDL for databasen:



4. Innhenting av bilder:

De første bildene ble hentet inn med GeoCam på Nokia N95, men siden programmet fortsatt var i utviklingsfasen, og det var tungvint å håndtere, ble de fleste fotografert med eget Casio Z1080 digitalkamera. På disse bildene ble GPS posisjonen hentet inn manuelt via Google Maps. Et problem var imidlertid at bildene fra kamera tok betydelig mer plass enn bildene fra mobiltelefonen (ca 4mb VS 100kb), så de ble redigert ned til 1024x768 i freeware fotoprogrammet Paint.NET.

Jeg hadde på forhånd plukket ut nye objekter som det skulle fotografertes, samt at jeg ville hente inn noen nye bilder av objektene som allerede eksisterte i BergenBy

databasen. Informasjon om objektene som ble hentet inn via google søk og fra wikipedia, samt tekstene til objektene fra wikipedia er satt til den personen som lastet ned og editerte teksten, der forfatteren av selve teksten var vanskelig å fremspore. Tekstene som er hentet fra turistinformasjonssider er derimot blitt registrert med korrekt forfatter. De historiske bildene som er hentet inn fra Universitetsbibliotekets samling er også registrert med den fotografen som tok bildet, samt omtrentlig dato. Det er også lagt inn noen bilder og objekter hentet inn av Joan Nordbotten.

Ved prosjektslutt har BergenBy databasen til sammen 250 bilder av 23 objekter, som er registrert med all metadata. Det er også hentet inn bilder som har flere objekter (opp til 3 objekter) i samme bilde, samt 8 bilder som ikke er knyttet til noen objekter som ikke er regnet med i listen. Disse er hovedsakelig testbilder som Lars-Jacob ønsket å ha med, og kan være interessante med tanke på CBIR søk. 4 av bildene er tatt om natten og er dermed mye mørkere enn de andre bildene, som også kan være interessant med tanke på bildesøk.

4.1 Bildeoversikt:

Tema					
Kategorier	Opptelling etter bilder som ligger under http://bulmeurt.uib.no/caim/images/ både UBB og CAIM-bilder er inkludert.				
Kirker 34	Domkirken 9	Johanneskirken 9	Korskirken 13	Fana Stavkirke 3	
Statuer 28	Chr. Michelsen 9	Grieg 8	Holbergstatuen 11		
Bygninger 45	Børsbygningen 16	Greghallen 11	Zachariasbryggen 12	Gamlehaugen 6	
Museer 53	Vestlandske Kunstindustrimuseum 10	Rosenkrantzårnet 18	Håkonshallen 20	Troldhaugen 5	
Monumenter 38	Sæverudmonumentet 12	Musikkpaviljongen 13	Sjømannsmonumentet 12	Rogaland Cross 1	
Annet 52	Festplassen 14	Lille Lungegårdsvannet 15	Beffen 13	Statsraad Lehmkuhl 10	
Totalt antall bilder 250					

5. Verktøy:

Verktøyene som ble benyttet i løpet av CAIM prosjektet:

- Dell Vostro 1500 Laptop.
- Microsoft Windows Vista Operativsystem.
- Mozilla Firefox 2.0.0.16
- Google Maps
- Eclipse 3.3.2
- Java version 6 update 7
- Oracle SQL Developer 1.5.1
- Paint.NET 3.31
- NetBeans IDE 6.1
- Nokia N95
- Casio Z1080

Eclipse ble benyttet til JAVA programmering av metadata editorene. Netbeans ble benyttet til selve eksporteringen av prosjektfilene for å opprette eksekverbare JAR filer av metadata editorene.

Oracle SQL developer ble benyttet til editering av kopiene til BergenBy databasen.

Nokia telefonen og Casio kamera ble benyttet for å hente inn bildene som ble fotografert i løpet av sommerprosjektet. Google Maps ble benyttet til å hente ut GPS koordinatene til bildene som ble fotografert med kamera, og Paint.NET brukt til å redigere kvaliteten til bildene slik at de tilsvarte filstørrelsen til mobilbildene.

Kun Windows Vista ble benyttet som operativsystem som kjørte de ulike verktøyene, og nettleseren Mozilla Firefox til Google Maps.

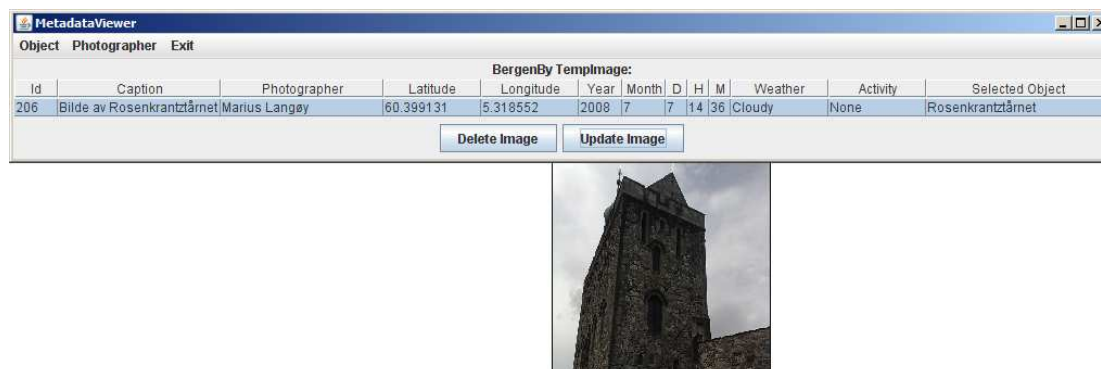
6. Metadata Editor Prototype

6.1 Videreutvikling av Maritim Editor?

Det eksisterte fra før av en metadata editor knyttet til Maritim databasen på bulmeurt som var utviklet i PHP. Det ble stilt forslag om videre utvikling av denne til å kunne fungere på BergenBy databasen ved prosjektstart, men på grunn av problemer knyttet til å få den til å fungere som ønsket, samt at gruppen hadde vesentlig mer erfaring med JAVA, ble det besluttet å lage en ny separat metadata editor til BergenBy i JAVA.

6.2 ImageEdit:

Denne prototypen er utviklet med intensjonen å endre metadata om bildene som eksisterer i Image tabellen i BergenBy.



6.2.1 Funksjonalitet:

ImageEdit har funksjonalitet for:

- Endring av caption til valgfri tekst.
- Endring av fotograf referanse til en av fotografene lagret i databasen (hentes ut som en liste).
- Endring av GPS breddegrad.
- Endring av GPS lengdegrad.
- Endring av dato (År, Måned, Dag, Time, Minutt).
- Endring av værforhold til valgfri tekst.
- Endring av aktivitet til valgfri tekst.
- Slette bilder fra databasen.
- Legge til nye objekter knyttet til bildet.
- Fjerne objekter knyttet til bildet.
- Legge til nye objekter i databasen.
- Editere eksisterende objekter i databasen.
- Slette objekter fra databasen.
- Legge til nye fotografer i databasen.
- Editere eksisterende fotografer i databasen.
- Slette fotografer fra databasen.

ImageEdit har ennå ikke funksjonalitet for:

- Legge til, editere og fjerne nøkkelord til bilder og objekter (nøkkelord er en nøstet tabell med referanser til nøkkelord).
- Legge til, editere og slette tekster til bilder og objekter (tekster er også en nøstet tabell med referanser til tekster, så denne løsningen kan være identisk til løsningen for nøkkelord).

6.2.2 Utvikling:

Metadata editoren er ennå kun en prototype, og selve koden er derfor ikke ennå delt opp og modularisert siden den fortsatt er under utvikling og stadig endring. Selve kodedelen til imageEdit og tempEdit er også nesten helt identisk, så det vil muligens kunne være hensiktsmessig å slå dem sammen til en editor, og la brukeren få velge om det er Image eller Image_Temp tabellen som skal kobles til. Kildekoden til imageEdit er kommentert, og ettersom tempEdit nesten benytter seg av identisk kode har det ikke blitt prioritert å legge kommentarer til denne ennå.

Begge editorene består av en JTable med cellene: "Id", "Caption", "Photographer", "Latitude", "Longitude", "Year", "Month", "D", "H", "M", "Weather", "Activity", "Selected Object". Tabellen er ca 1000 pixler bred og vil derfor ikke kunne vises på skjermopløsninger som er under 1024 x 768.

"Id" er tilknyttet en JComboBox som henter ut alle Image ID som ligger i Image tabellen (Image_Temp tabellen for tempEdit). De andre cellene tar grunnlag i denne Id'en når programmet henter ut metadata om bildet fra databasen.

Image:

Et av de større problemene i utviklingen var å få editoren til å hente ut selve ORDIMAGE bildet fra databasen og vise det til brukeren. Løsningen som editoren bruker nå er å hente ut

```
String spørring = "SELECT Image from Image where id="+id;
```

som et OracleResultSet.

```
OracleResultSet rs = (OracleResultSet) stmt.executeQuery(spørring);
```

Deretter hentes bildet ut fra resultsettet som et OrdImage:

```
OrdImage imageProxy = (OrdImage) rs.getORADData("image",  
OrdImage.getORADDataFactory());
```

En `InputStream` lages fra `OrdImage`.

```
InputStream is = imageProxy.getContent().getBinaryStream();
```

En `ImageInputStream` av `InputStream`.

```
ImageInputStream iis = ImageIO.createImageInputStream(is);
```

`BufferedImage` fra `ImageInputStream`.

```
BufferedImage image = ImageIO.read(iis);
```

Skaleres ned til et `Image`.

```
Image img = image.getScaledInstance(200, 200, Image.SCALE_FAST);
```

Dette er en tungvint prosess som muligens kan gjøres enklere, og nok er grunnen til at programmet i begynnelsen ofte fikk "Out of Memory" error. Selve bildet plasseres i et egen `JWindow` plassert under vinduet som tabellen er i.

Photographer/Selected Object:

I likhet med `Image Id` hentes navnet på alle fotografer og objekter fra databasen, og legges til egne `JComboBox`er knyttet til disse cellene. Brukeren vil dermed ha mulighet til å velge fotograf og object fra en dropdown meny.

Delete Image:

Knappen for "Delete Image" fungerer ved at den henter den vagte `Image Id` fra tabellen, og sletter så bildet fra `Conatins` tabellen i tilfelle det er tilknyttet noen objekter, og så fra `Image` tabellen. Til slutt fjernes den valgte `Id`'en fra `Image ComboBox`en .

```
dbfacade.delete("delete from contains where deref(image).id="+id);  
dbfacade.delete("delete from image where id="+id);  
imageComboBox.removeItem(id);
```

Update Image:

Knappen for "Update Image" leser inn alle de nye verdiene fra JTable (og legger til default verdier dersom de skulle være null). Den gjør så SQL spørringer for å legge til de nye metadata til bildet. Den sjekker så om det er valgt et object. Hvis det er et valgt object og det ikke eksisterer fra før knyttet til det valgte bildet i Contains tabellen, lages det en ny relasjon mellom bildet og objektet ved en Insert i Contains tabellen. Hvis referansen mellom bilde og object allerede eksisterer, eller ikke noe object er valgt, gjør den ikke noe.

Object:

Add New Object: Gir brukeren mulighet til å opprette et nytt Object gjennom flere `JOptionPane.showInputDialog` valg. Hvis noen av valgene er tomme, settes det på default verdier.

Edit Selected Object: Lager en ny JFrame med en egen JTable som henter ut alle verdiene til objectet som er valgt til Image. Framen har knapper for å "Cancel" og "Update", som henter ut alle verdiene som er satt inn i tabellen, og gjør spørring for å oppdatere objectet.

Remove Referance to Selected Object: Henter ut ImageId og Id på det valgte objectet, og fjerner fra Contains der disse eksisterer:

```
dbfacade.delete("DELETE FROM CONTAINS WHERE  
DEREF (IMAGE) .ID="+selectedImageId+" AND  
DEREF (OBJECT) .ID="+selectedObjectId);
```

Delete Selected Object: Henter Id på valgt object, og sletter fra Contains og Object der dette finnes, og fjerner det fra ComboBox.

```
dbfacade.delete("DELETE FROM CONTAINS WHERE  
DEREF (OBJECT) .ID="+selectedObjectId);  
dbfacade.delete("DELETE FROM OBJECT WHERE ID="+selectedObjectId);  
objectComboBox.removeItem(objectName);
```

BergenBy DB

Photographer:

Add New Photographer: Gir brukeren mulighet til å opprette en ny fotograf, på samme måten som med object.

Edit Selected Photographer: Gir brukeren mulighet til å editere en valgt fotograf, på samme måten som med object.

Delete Selected Photographer: Gir brukeren mulighet til å slette en valgt fotograf, på samme måten som med object.

Exit: Må brukes til å avslutte programmet, ettersom X-funksjonen i selve vinduet er slått av.

6.2.3 Kjente bugs:

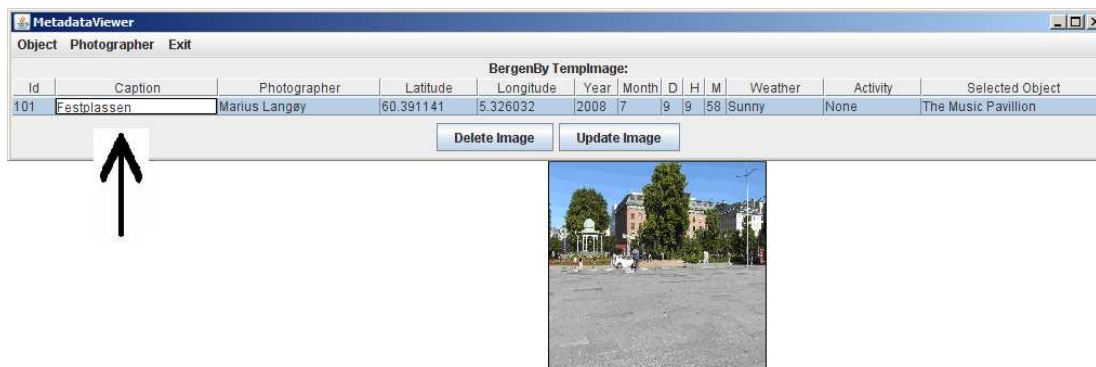
Combobox: Dersom det velges samme object eller fotograf 2 ganger på rad, vil comboboxen stå i en slags mellomstilling, og valget vil ikke komme med på `.getselectedValue()`. Dette kan løses ved å velge et annet object, og deretter trykke tilbake til det opprinnelige igjen, men er en irriterende feil for brukeren.

Illustrasjon:



JTable: Dersom en av cellene i tabellen fortsatt er aktiv (dvs. en ikke har trykket enter ennå) når en velger ”Update Image”, vil ikke noe av informasjonen i cellen bli registrert og det vil bli lagret som ”Not set”.

Illustrasjon:



Out of Memory: Editorene viser relativt små versjoner av bildene det er metadata om (200x200 pixler). Grunnen til dette er at hvis de vises større vil Java gi feilmelding om Out of Memory. Dette tyder på at det er en minnelekasje et sted i programmet ved tanke på visning av bildene.

SQL Injections: Det skal være feilsjekker som forhindrer SQL injections i metadata editoren.

Feil input: Editorene har ennå ikke noen feilsjekk for at datatypene som skrives inn i feltene er korrekte, f.eks at tallene for dato stemmer, og at GPS koordinatene har riktig format. Dette vil enkelt kunne ordnes ved å putte inn if-sjekker for slik inndata.

7. Future development/tasks

- Framtidig utvikling av editoren vil være å legge til funksjonalitet for å editere tekster og nøkkelord for bildene og objectene, samt å sørge for at editoren ikke kan motta ugyldig input i feltene for dato og koordinater. Det kan også være hensiktsmessig å se nærmere på hvorvidt det eksisterer en enklere måte å displaye bildene fra databasen i editorene.
- Framtidig utvikling av databasen kan være i form av å utvide metadata til å inkludere lydfiler fra objektene som spilles av på mobiltelefonen, eller TEXT-TO-SPEECH funksjoner for opplesning av tekstene til hvert bilde og object.
- Med tanke på GPS er dette lagret som et punkt i SDO_GEOMETRY form i databasen både for bilder og objecter. For bilder vil dette være hensiktsmessig for å vise hvor personen som fotograferte bildet befant seg, men det kan være en mindre god måte å vise hvor objectene befinner seg siden det vil være vanskelig å si nettopp hvor det sentrale punktet i objectet er. Dette kan utbedres ved at GPS lokasjonen til objectene heller lagres som en figur av GPS punkt, noe som skal være støttet av SDO_GEOMETRY.

8. User manual, with screen shots

Filer som utgjør programmet:

ImageEdit.jar – Eksekverbar fil for å starte bildeeditoren.

TempEdit.jar – Eksekverbar fil for å starte editoren som flytter et bilde fra Image_Temp til Image tabellen.

Lib – Mappe for eksterne jar-filer.

ojdbc14.jar – Ekstern jar-fil for databasetilkobling.

ordim.jar – Ekstern jar-fil for bildehåndtering.

runtime12.jar – Ekstern jar-fil for databasetilkobling.

8.1 MetadataEditors:

8.1.1 TempEdit:

Metadata editoren vil gi en oversikt over alle bildene som ligger i Image_Temp tabellen på BergenBy. Dette vil altså være de bildene som har blitt tatt med kameratelefonen. Disse bildene vil derfor kun inneholde metadata om dato og GPS. Hensikten med metadata editoren er derfor å tilføre flere metadata til bildene og flytte dem over i Image tabellen.

Metadata som tilføres av editoren er:

Id: Blir automatisk lagt inn av en prosedyre.

Caption: Tittelen på bildet. Skrives inn manuelt i feltet.

Photographer: Fotograf for bildet. Velges ut ifra en dropdown-meny. (OBS. den har ennå ikke funksjonalitet for å opprette nye fotografer i databasen).

Coordinates: Longitude og Latitude. Skal legges inn automatisk av kameramobil, men kan editeres dersom ønskelig.

Date: År, Måned, Dag, Time og Minutt. Skal legges inn automatisk av kameramobil, men kan editeres dersom ønskelig.

Weather: Værforhold på bildet. Skrives inn manuelt i feltet. Det kan tenkes at fremtidige kamera vil ha mulighet til å automatisk sende data om værforhold.

Activity. Eventuell aktivitet i bildet. Skrives inn manuelt i feltet.

Object: Referanse til objektet det tas bilde av. Velges ut ifra en dropdown-meny. Metadata editoren kan også opprette nye objekter i databasen dersom det ikke eksisterer fra før.

Begrensinger:

Det eksisterer også metadata i form av tekst knyttet til bildet, og nøkkelord som beskriver bildet i databasen, men metadata editoren har ennå ikke støtte for å kunne legge inn dette.

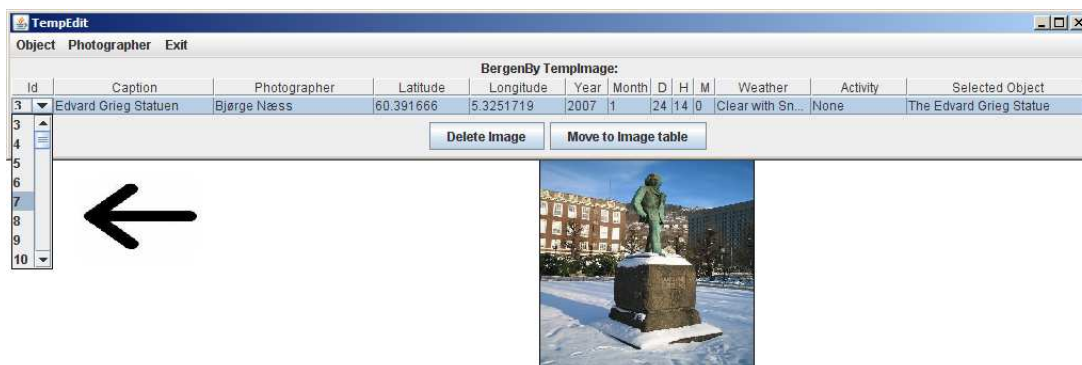
Installasjon av editoren:

Metadata editoren startes ved å pakke ut .rar filen den er lagret i, og gjennom filen 'TempEdit.jar'. Det er viktig at mappen 'lib' med de eksterne filene som benyttes ikke flyttes.

Bruk av Editoren:

Når den starter skal metadata editoren automatisk hente inn ID til alle bildene som eksisterer i Image_Temp. Den viser metadata som er knyttet til dette, samt et vindu med bildet.

En velger hvilket bilde en vil flytte over til Image tabellen ved å trykke på ID cellen. En dropdown meny over alle de eksisterende ID'ene vil deretter la en navigere seg gjennom de ulike bildene.

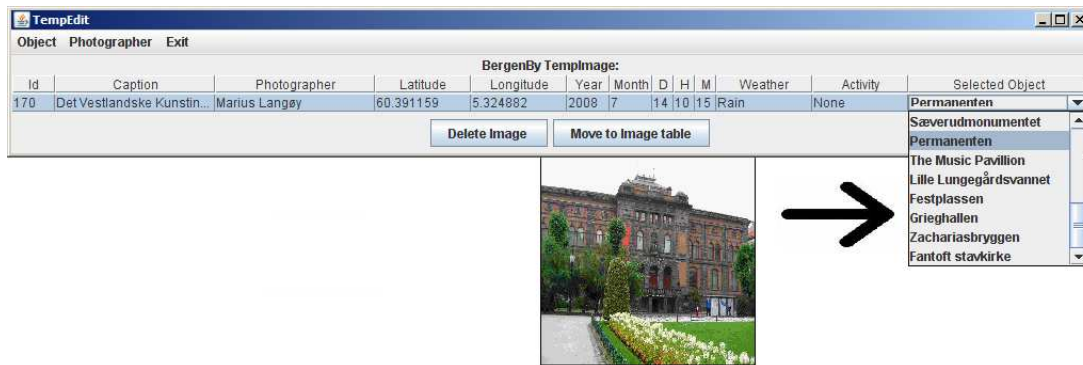


Når en har komt fram til bildet en ønsker å flytte over, velger en fotograf ved å trykke på Photographer cellen. En vil da få mulighet til å velge mellom alle fotografene som er registrert i databasen.

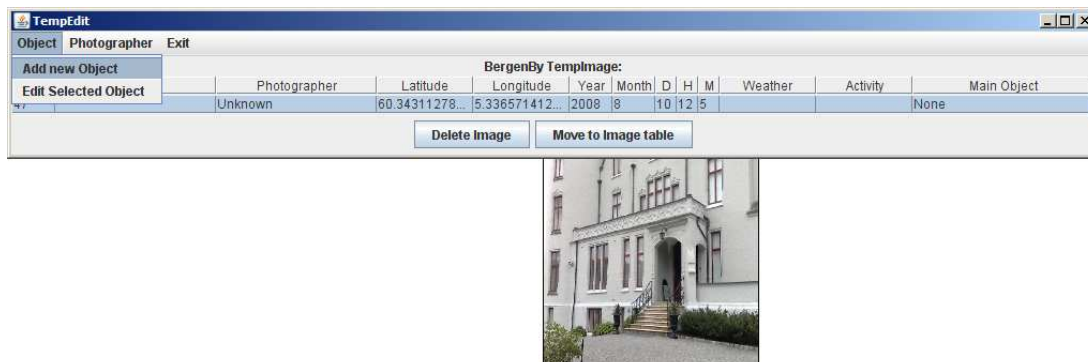


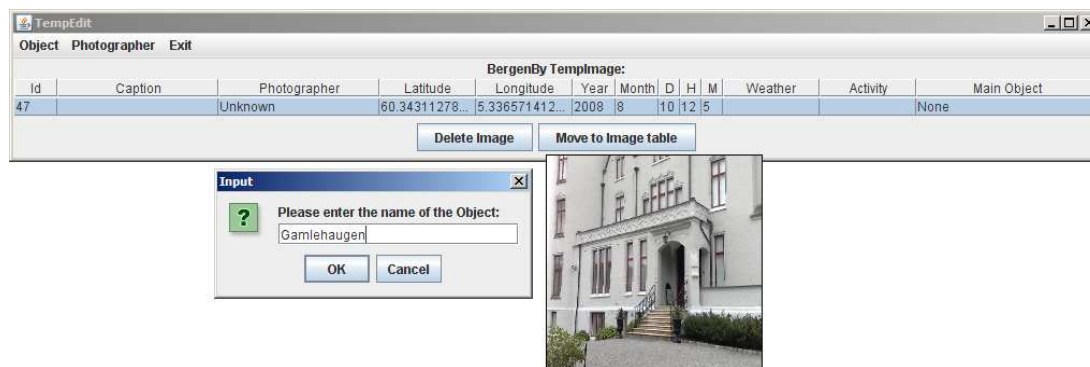
En fyller så ut de ytterlige metadata ved å trykke på cellene og skrive inn ønsket verdi. Programmet vil ignorere spesialtegnet for apostrof ('), for å forhindre SQL injections. Editoren vil sette default verdier dersom noen av cellene er tomme, men det er viktig å være obs på at en ikke må ta inn ugyldige verdier som f.eks bokstaver i dato eller GPS-feltene ettersom den ennå ikke inneholder feilsjekk for dette. Det er også mulig at det eksisterer sikkerhetshull for SQL-injections.

TempEdit har for øyeblikket bare støtte for å knytte ett objekt opp mot bildet som det legger inn i Image tabellen. (Hvis det skal knyttes flere objekter til bildet må dette gjøres i ImageEdit etter at bildet er lagt til). Objektet velges ut ifra en dropdown meny i objekt cellen. Denne henter inn en liste over alle objektene som eksisterer i databasen.

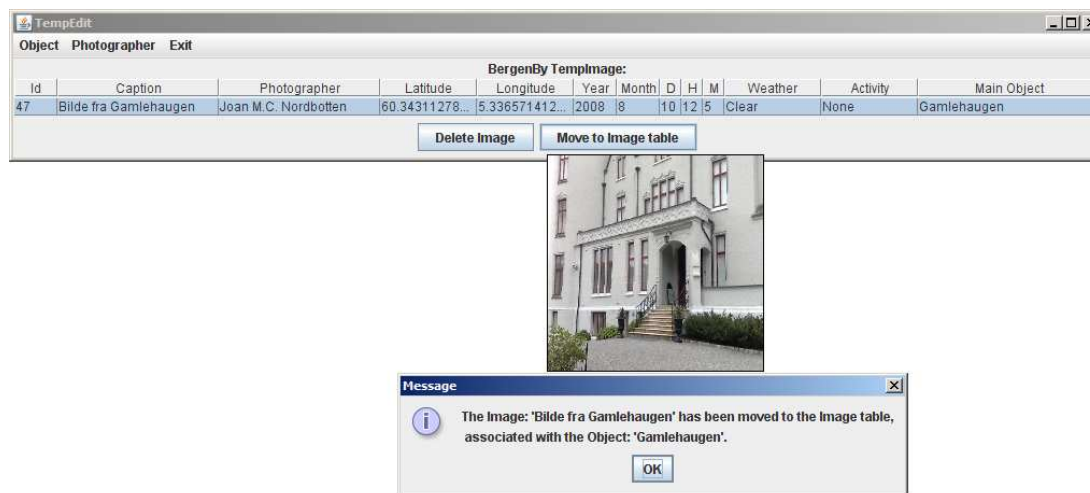


Dersom bildet tilhører et objekt som ikke finnes i databasen, vil en kunne opprette et nytt objekt ved å velge Object – Add New Object fra menyen øverst til venstre. Denne menyen vil spørre brukeren etter de nødvendige data for å opprette et nytt objekt. Dersom feltene er tomme vil den legge til default verdier, men en må også her være obs på å ikke gi editoren ugyldige verdier i form av bokstaver i tallfelt.





Når all metadata er gitt, vil en kunne flytte over bildet til image tabellen ved å trykke på knappen ”Move to Image Table”. Dersom en valgte et objekt, vil editoren gi beskjed om tilknytningen.



Avslutte editoren:

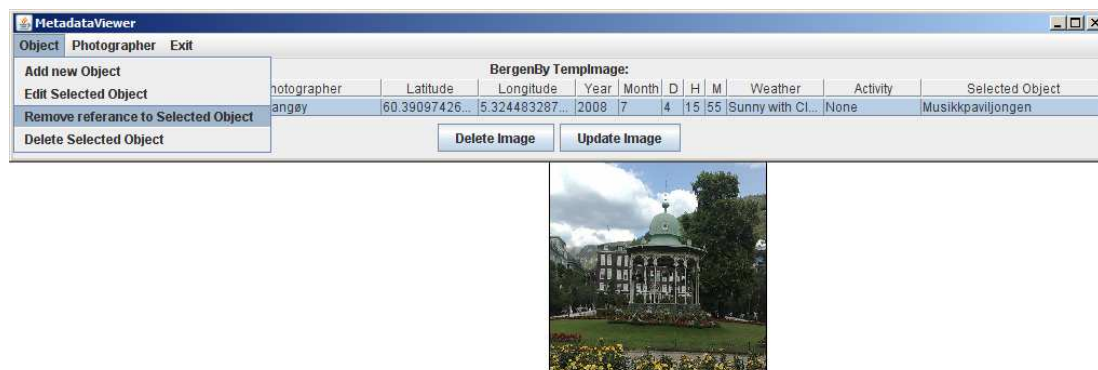
Hvis editoren avsluttes med standard windows ”x” knapp øverst til høyre i vinduet, vil ikke alltid JAVA lukke vinduet som inneholder bildet. Denne funksjonen er derfor avslått, og programmet bør avsluttes via hovedmenyen Exit – Quit.

8.1.2 ImageEdit:

ImageEdit er stort sett akkurat det samme programmet som TempEdit, men henter ut bildene fra Image tabellen i stedet for Image_Temp. Dette er fordi at denne editoren skal brukes til å editere metadata knyttet til selve bildene som er lagt inn i databasen. Knappen for "Move to Image Table" er derfor i stedet kalt "Update Image", og oppdaterer databasen med de nye metadata ved gjeldene ImageId.

En svakhet med ImageEdit er at den kun viser brukeren ett av objektene som hvert bilde er assosiert med, men det er mulighet for å knytte flere objecter til ett bilde. Dette gjøres ved å velge et nytt object fra object-menyen, og deretter "Update Image". Bildet vil da bli knyttet til det nye valgte objectet i tillegg til det gamle.

Funksjonalitet som er unik for ImageEdit er valgmuligheten under Object som er kalt "Remove Referance to Selected Object".



Editoren vil da sjekke om bildet er knyttet til det valgte objectet. Hvis det er det, fjernes denne referansen. Hvis ikke, får brukeren tilbakemelding om at de ikke er knyttet sammen.

ImageEdit har også funksjonalitet for å slette valgt Object og Photographer under de tilsvarende menyene. Hvis fotograf slettes vil bildene som har den valgte fotografen få null-verdier i photographer. Hvis Object slettes, vil alle referansene mellom bilder

BergenBy DB

og det valgte objectet fjernes slik at det ikke blir null-verdier i Contains tabellen, før objectet slettes.

Foruten om disse funksjonene fungerer programmet på samme måte som TempEdit.

References

Appendix:

1. DB data
 - All databasestruktur, insert-setninger for all data, kildekode for metadata editorene samt originalversjonen av alle bildene ligger på vedlagt CD.