

# Rapport semesteroppgave INFO216

Vår 2018

Kandidatnummer: 106, 120 og 123

Denne oppgaven har resultert i en nettside som presenterer ti åpne datasett om Gjesdal kommune. Her kan man filtrere etter spesifikt datasett, finne avstand fra et bestemt punkt til alle objekter i et valgt datasett, eller skrive sin egen spørring. Vi valgte å fokusere på Gjesdal kommune, fordi de har mange flere åpne datasett enn andre kommuner. Av de 15 åpne datasettene Gjesdal kommune har, valgte vi å bruke 10 av dem, fordi det var de som var mest interessant å koble sammen. De valgte datasettene omhandler returpunkt, badeplasser, skoler, barnehager, fiskeplasser, grillplasser, kirker, overnatting, skjenkesteder og utleielokaler. Datasettene var i utgangspunktet ikke koblet sammen på noen måte, så vi synes det var en god idé å koble de til hverandre, ved hjelp av semantiske teknologier, for å kunne se sammenhengen mellom dem. På den måten kunne vi lage en nettside som presenterer dataene, hvor man kan søke og filtrere på kryss av datasettene, ved å bruke SPARQL.

Alle datasettene inneholder lengdegrad og breddegrad, så vi har valgt å bruke disse for å finne avstand mellom de forskjellige stedene. Dette gjorde det naturlig for oss å bruke et kart for å presentere dataene. Når man trykker på et punkt på kartet får man opp navnet på stedet. Hvis dette objektet har en beskrivelse i datasettet vil denne også vises i boksen som dukker opp. Vi viser også resultatene i tabell og som tekst. I flytdiagrammet (figur 1 vedlagt) kan man se nærmere på hva man kan gjøre på nettsiden.

Vi valgte å lage vår egen ontologi for dette prosjektet da vi ikke fant en eksisterende ontologi som passet godt nok. For å lage ontologien brukte vi Protégé. Først lagde vi klasser for de forskjellige datasettene, en klasse for kirker, en for grillplasser osv. Vi så etter hvert at mange av klassene i ontologien vår var ekvivalente til eksisterende klasser i schema.org, derfor endret vi på dette slik at vi bruker noen klasser fra Schema.org, i stedet for å opprette nye. Vi fant ingen passende klasser på schema.org for grillplasser og returpunkt. Du kan se en tegning av ontologien i figur 2 i vedlegget.

Vi lastet ned datasettene i CSV-format fra data.norge.no. For å kunne importere dataene inn i ontologien måtte vi først endre formatet slik at de var i XLSX-format. Noen av datasettene hadde gatenavn, gatenummer, postnummer og poststed i samme kolonne. Vi ville lagre disse dataene som forskjellige egenskaper, så vi måtte dele opp kolonnene slik at det ble lettere å importere de i ontologien. Ideelt sett skulle vi ikke måtte endre datasettene, men vi valgte å gjøre dette for å kunne presentere de på best mulig måte og for at man i fremtiden skal kunne koble sammen stedene basert på lik adresse. Etter dette importerte vi datasettene inn i Protégé ved å lage regler ved hjelp av Manchesternotasjon for hvert av datasettene. Reglene ble lagret i egne filer, dette gjorde det mye enklere for oss å gjøre endringer i ontologien og å importere datasettene på nytt.

Vi brukte Eclipse for å lage en klasse (Model.java) som leser inn ontologien og gjør den om til en Turtle-fil (Gjesdal.ttl). Før vi genererer Turtle-filen, vil vi legge til litt mer informasjon til ontologien. For noen av stedene i datasettene (Gjesdal kommune, kirker, noen fiskeplasser, og én bar) finnes det mer informasjon på DBpedia.org. Dette valgte vi å inkludere, for å gjøre nettsiden mer innholdsrik, da datasettene ikke inneholder så mye informasjon. Vi valgte å inkludere dbo:Abstract til disse stedene da de beskriver lokasjonen i tillegg til litt generell informasjon. På DBpedia finnes det mye annen informasjon man kan inkludere, men det var denne vi fant mest interessant. Når man søker på en av stedene som har ekstra informasjon i nettsiden, vil denne informasjonen bli synlig under kartet og/eller tabellen. Informasjonen om Gjesdal kommune vises på toppen av nettsiden.

Model.java henter altså ut dbo:Abstract-elementet til de stedene vi fant på DBpedia og legger dem til på riktig ressurs. For å hente elementet kjøres funksjonen getAbstract() fra DBpedia.java som tar inn en URL fra DBpedia og henter ut dbo:Abstract elementet. Filen som Model.java lager må vi så manuelt laste opp på Fuseki-serveren for at nettsiden skal kunne sende spørringer og vise brukeren resultatene av spørringene.

På den delen av nettsiden hvor man kan skrive inn en egen spørring valgte vi å legge ved en link til en side som viser datasettet, slik at brukeren selv kan se på settet og finne stedsnavn, adresse, og lignende. Dette gjør det lettere for brukeren å skrive spennende spørringer.

I prosjektet har vi brukt GitHub for versjonskontroll. Her har vi organisert prosjektet med mapper for å holde oversikt over de forskjellige filene og vi har opprettet egne brancher når vi prøver ut nye ting. Vi har brukt programmet Atom for å skrive HTML-, CSS- og JavaScriptkode til nettsiden. Vi brukte dette fordi det har en plug-in kalt Teletype som lar oss skrive i de samme dokumentene samtidig. Dette lar oss enklere jobbe sammen og har gitt oss en bedre forståelse for hverandre sin kode. Vi har for det meste jobbet sammen på lab og i spørretimene og har alle jobbet og bidratt med de forskjellige delene av koden og oppgaven.

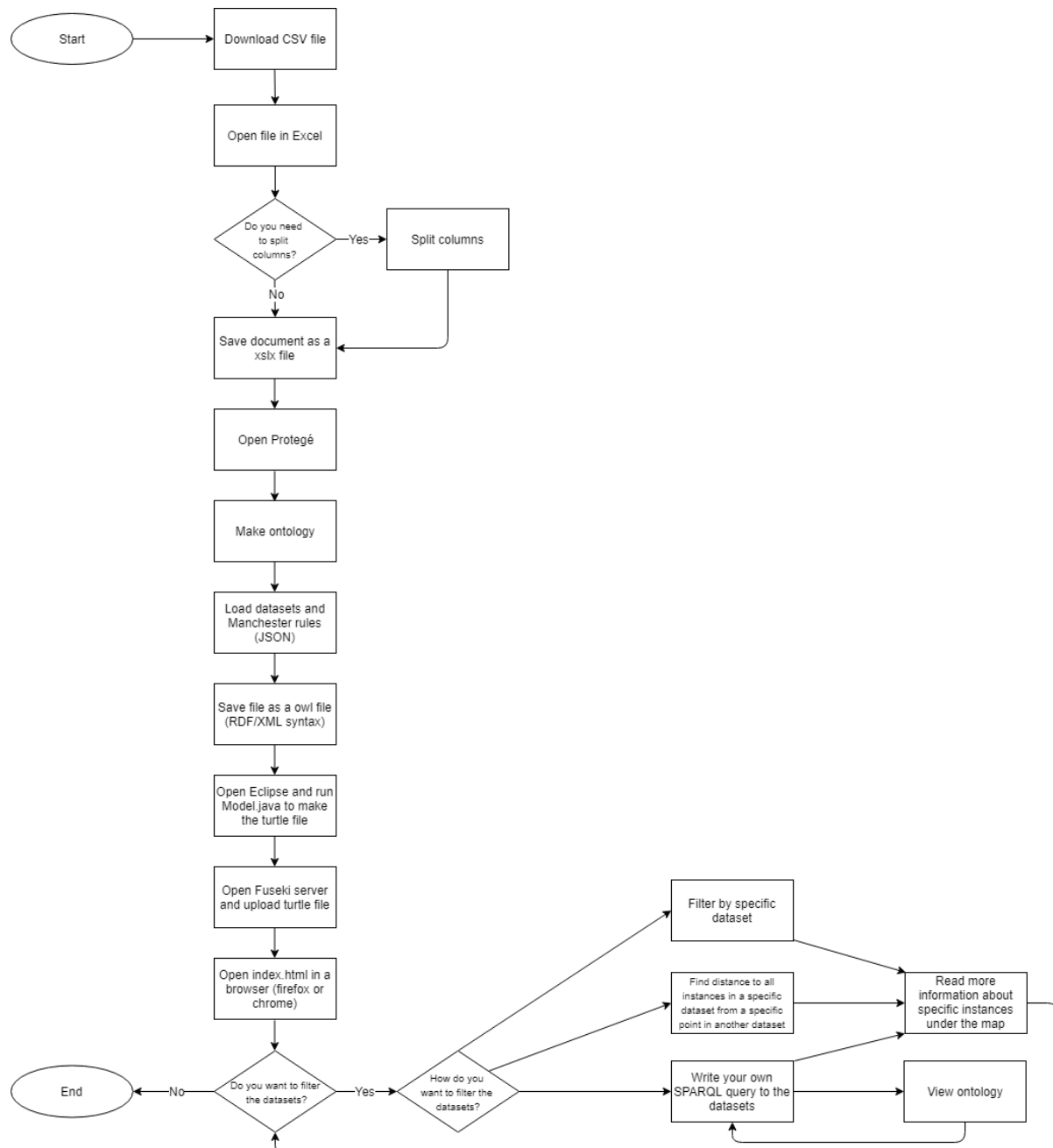
Vi prøvde å få GeoSPARQL inn i ontologien for å enklere kunne regne ut avstander mellom de forskjellige stedene i datasettene. Det viste seg at om vi skulle få GeoSPARQL til å fungere sammen med Fuseki-serveren måtte vi ha skrevet eller importert mye Javakode da Fuseki ikke uten videre støtter GeoSPARQL. Vi oppdaget GeoSPARQL på et sent tidspunkt i oppgaven og ettersom vi allerede hadde skrevet en komplisert spørring som regner ut avstanden, fant vi ut at vi like godt kunne bruke denne. Hadde vi oppdaget dette på et tidligere tidspunkt ville vi prøvd å inkludere GeoSPARQL.

I prosjektmappen ligger det tre mapper for å organisere filene til prosjektet. Her ligger også README.txt som beskriver hva som skal til for å få nettsiden/programmet til å kjøre. I mappen Datasett ligger alle datasettene som XLSX-filer og en mappe, Regler, hvor reglene vi lagde i Manchester-notasjon til hvert datasett er lagret som JSON filer. I mappen «Gjesdal kommune» ligger Java-prosjektet vårt. Inne i denne mappen ligger også Gjesdal.ttl filen (Turtle filen som blir laget/overskrevet i Java-koden). Under mappen "prosjekt" i src-mappen ligger de to Java-filene vi bruker i prosjektet, i tillegg til Ontologi.owl filen som er ontologi-filen vår. Under mappen Nettside ligger alt vi bruker for å vise frem nettsiden: HTML, CSS, JavaScript og fonten vi bruker. Da det ikke er så mange filer har vi valgt å ha de i samme mappe.

Vi valgte å ta utgangspunkt i Gjesdal kommune fordi de har mye åpne data tilgjengelig. Vi tror at systemet vårt, med litt tilpasning, kan overføres til andre kommuner eller fylker, dersom de publiserer åpne data om lokasjoner i fremtiden.

## Vedlegg:

**Figur 1:** Flytdiagram (også vedlagt i prosjektmappen under navnet FlowChart\_Gjesdal.png)



**Figur 2:** Ontologidiagram (også vedlagt i prosjektmappen under navnet Ontologidiagram.png)

