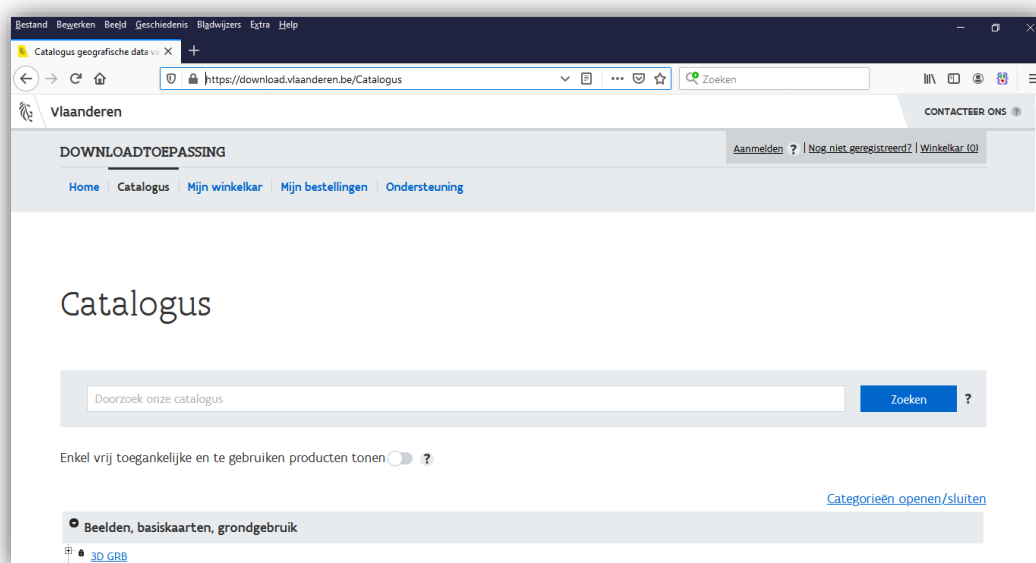


Adresbeheer

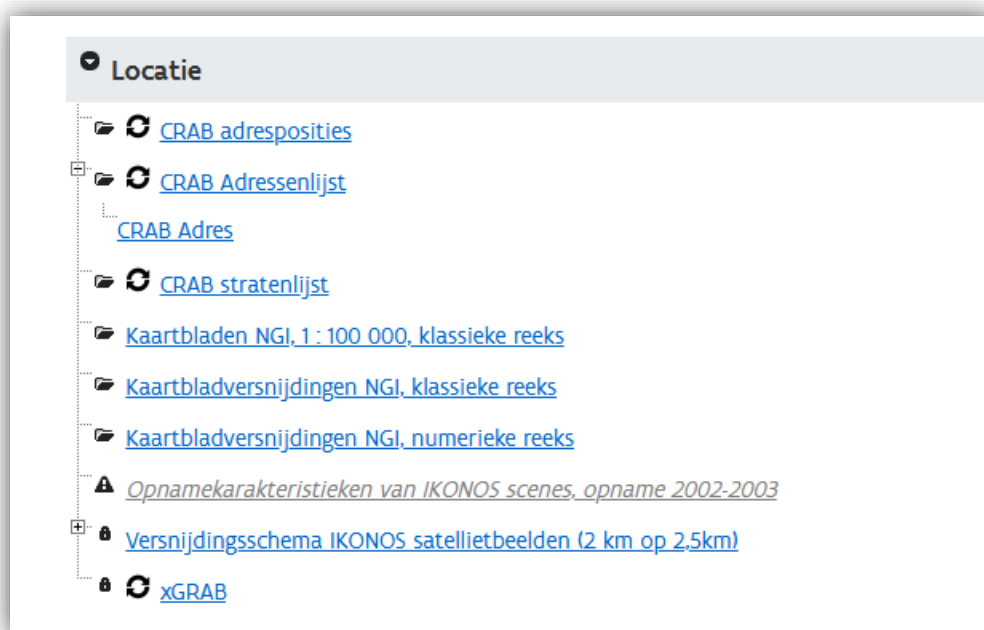
Downloaden adresdata

Op onderstaande link kunnen heel veel datasets gevonden worden die door de Vlaamse overheid ter beschikking worden gesteld.

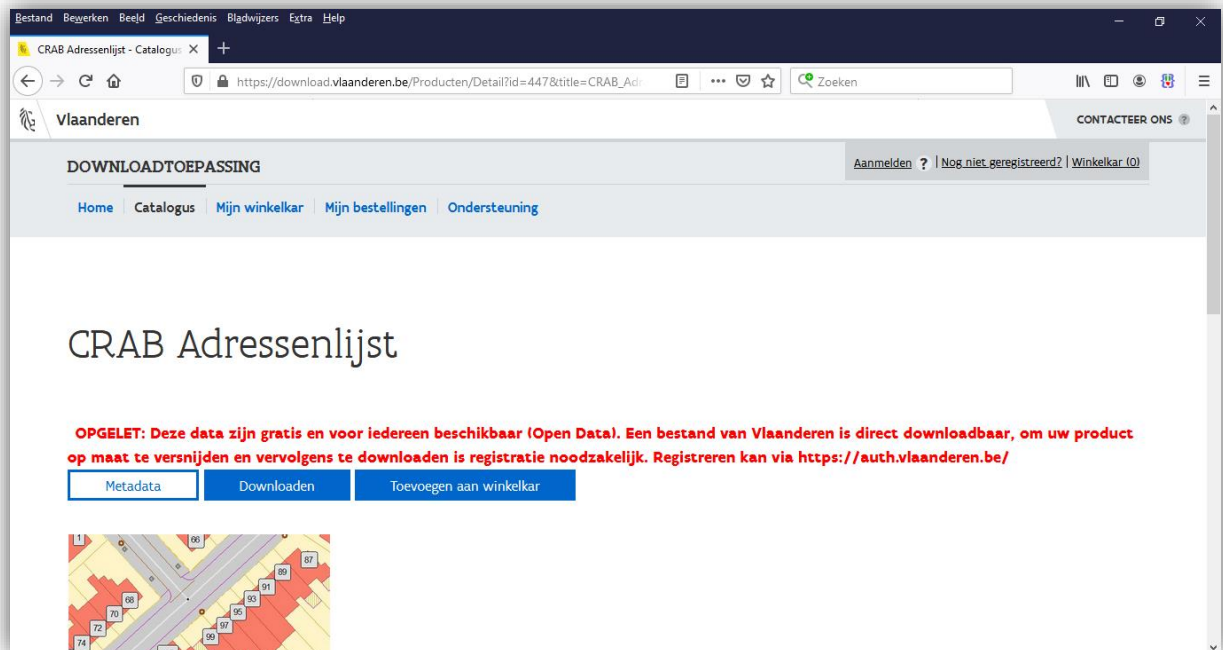
<https://download.vlaanderen.be/Catalogus>



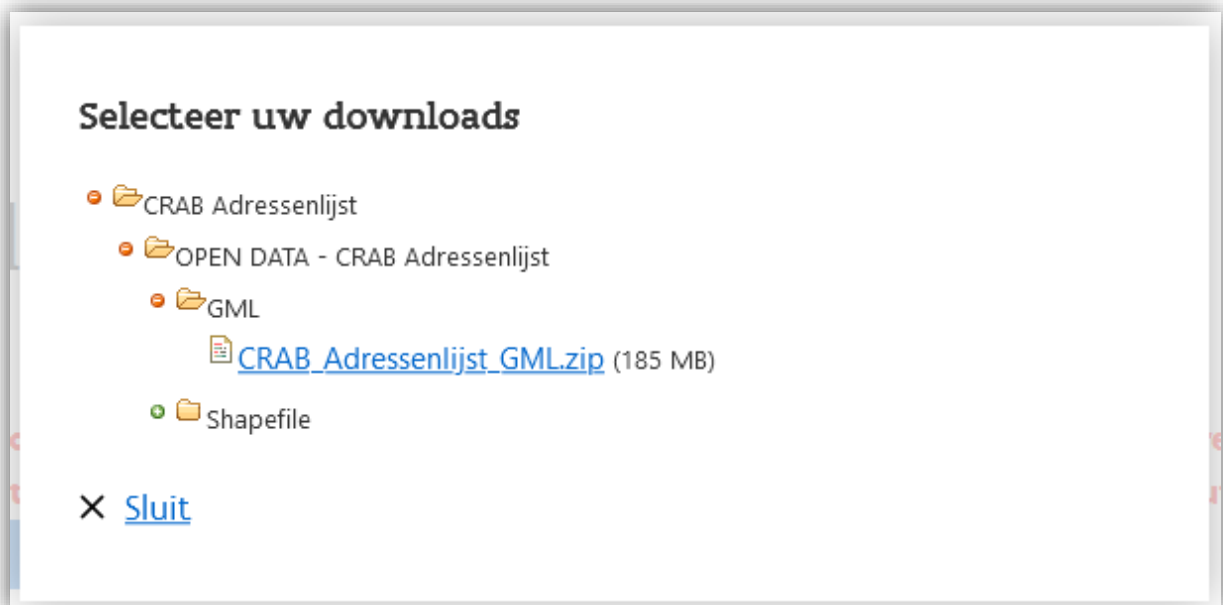
We zijn op zoek naar adresgegevens en die vinden we in de groep “Locatie” in de folder “CRAB Adressenlijst”. CRAB staat voor Centraal Referentie Adressen Bestand”.









Na het aanklikken van de link CRAB Adressenlijst komen we op de volgende pagina terecht, waarna we kiezen voor Downloaden.






Er worden verschillende formaten aangeboden, wij kiezen voor GML. GML staat voor Geography Markup Language (https://nl.wikipedia.org/wiki/Geography_Markup_Language) en heeft een XML codering.



Na het downloaden kunnen we het bestand unzippen. De data zelf vinden we in de folder GML, namelijk het bestand CrabAdr.gml. Dit bestand bevat de adresgegevens voor Vlaanderen.

	GML	File folder	
	Gebruik_CRAB_Adressenlijst.pdf	Adobe Acrobat Document	32 KB
	Leesmij_CRADRLST.pdf	Adobe Acrobat Document	821 KB
	Meldingsformulier.rtf	Rich Text Format	9 KB
	Meta_CRAB_Adressenlijst.pdf	Adobe Acrobat Document	209 KB
	Meta_CRAB_Adressenlijst.xml	XML Document	7 KB

	extra	File folder	
	CrabAdr.gml	GML File	189 330 KB No
	CrabAdr.xsd	XML Schema File	1 KB No

Bekijken we de inhoud van het bestand dan zien we de volgende structuur :

```

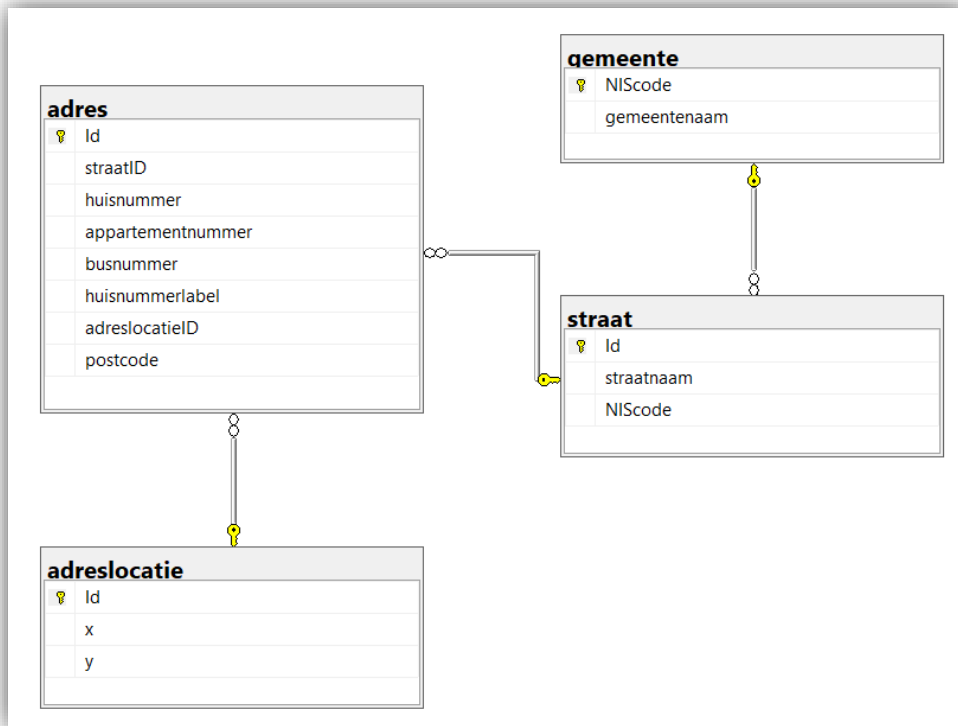
D:\NET\Tutorial\dataAdressen>type CrabAdr.gml | more
<?xml version="1.0" encoding="utf-8"?>
<agiv:FeatureCollection xmlns:agiv="http://www.agiv.be/agiv" xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="CrabAdr.xsd">
  <gml:boundedBy>
    <gml:Box srsName="EPSG:31370">
      <gml:coordinates />
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <agiv:CrabAdr>
      <agiv:ID>2000000038</agiv:ID>
      <agiv:STRAATNMID>77461</agiv:STRAATNMID>
      <agiv:STRAATNM>Polenlaan</agiv:STRAATNM>
      <agiv:HUISNR>50</agiv:HUISNR>
      <agiv:APPTNR />
      <agiv:BUSNR />
      <agiv:HNRLABEL>50</agiv:HNRLABEL>
      <agiv:NISCODE>46020</agiv:NISCODE>
      <agiv:GEMEENTE>Sint-Gillis-Waas</agiv:GEMEENTE>
      <agiv:POSTCODE>9170</agiv:POSTCODE>
      <agiv:HERKOMST>manueleAanduidingVanGebouw</agiv:HERKOMST>
      <gml:pointProperty>
        <gml:Point srsName="EPSG:31370" xmlns:gml="http://www.opengis.net/gml">
          <gml:coord>
            <gml:X>129968.00</gml:X>
            <gml:Y>215880.00</gml:Y>
          </gml:coord>
        </gml:Point>
      </gml:pointProperty>
    </agiv:CrabAdr>
  </gml:featureMember>
  <gml:featureMember>
    <agiv:CrabAdr>
      <agiv:ID>2000000039</agiv:ID>
      <agiv:STRAATNMID>77361</agiv:STRAATNMID>
      <agiv:STRAATNM>Drieske Nijperstraat</agiv:STRAATNM>
    </agiv:CrabAdr>
  </gml:featureMember>
</agiv:FeatureCollection>

```

We kunnen het bestand als een tekstbestand lezen, telkens we de string "<agiv:CrabAdr>" tegenkomen, hebben we te maken met een nieuw adres. De volgende lijnen bevatten dan telkens de eigenschappen van het adres. In het document over het datamodel beschrijven we welke attributen we zullen gebruiken.

Data model

Het databankmodel dat we gaan gebruiken ziet er als volgt uit :



Identificatoren

We hergebruiken de IDs uit het bestand dat we ontvangen. Voor adreslocatie is er geen ID meegegeven dus deze moeten we zelf genereren.

Implementatie

- We ontwikkelen in c# en .NET Core en maken gebruik van Visual Studio.
- We maken gebruik van ADO.NET voor de toegang tot de databank.
- Als databank gebruiken we Microsofts SQL Server.
- Voor de user interface maken we gebruik van WPF.
- Voor unit tests maken we gebruik van xUnit.

Gevraagde applicaties

1. Bouw een eenvoudige applicatie waarmee we de databank kunnen initialiseren met de gedownload data. Dit mag een console app zijn. Zorg ervoor dat het uploaden efficiënt gebeurt (richttijd : enkele minuten). Data die niet voldoet moet worden gelogd – geef ook mee wat het probleem is en welke data dit probleem veroorzaakt.
2. Plaats de business logica in een class library.
3. Bouw een gebruiksvriendelijke databeheer applicatie waarmee we de adressen kunnen beheren. Maak hierbij gebruik van WPF om de user interface te maken.

Databeheer

Model

Gemeente

- De naam van de gemeente mag niet leeg zijn.
- De naam van de gemeente begint met een hoofdletter.
- De NIScode moet een getal zijn dat bestaat uit 5 cijfers (en het eerste cijfer is niet 0).
[<https://nl.wikipedia.org/wiki/NIS-code>]

Straat

- De naam van een straat mag niet leeg zijn.
- De id van de straat moet een positief getal zijn.
- Een straat moet steeds naar een gemeente verwijzen.

Adreslocatie

- De id moet een positief getal zijn.
- De x-coördinaat moet groter zijn dan 22000 en kleiner dan 258000.
- De y-coördinaat moet groter zijn dan 152000 en kleiner dan 244000.

Adres

- De id moet een positief getal zijn.
- Het huisnummer mag niet leeg zijn en moet beginnen met een cijfer.
- De postcode moet bestaan uit 4 cijfers.
- Elk adres moet toebehoren aan een straat.
- Elk adres moet een locatie hebben.

User stories

In de beheerapplicatie moeten we in staat zijn om :

- Een gemeente toe te voegen, te verwijderen en te updaten. Zorg ervoor dat er geen dubbels voorkomen en dat geen gemeente wordt verwijderd die nog straten bevat.
- Een straat toe te voegen, te verwijderen en te updaten. Ook hier zorgen we ervoor dat een straatnaam uniek is in een gemeente en dat we geen straten verwijderen waar adressen aan zijn gekoppeld.
- Een adres toe te voegen, te verwijderen en te updaten. Dubbels zijn ook hier niet toegestaan.

In de user interface zullen we de gebruiker begeleiden/helpen door dropdownlijstjes van o.a. de gemeenten en straatnamen (afhankelijk van de noden). Om dit allemaal mogelijk te maken voorzien we een klasse die minstens de volgende interface implementeert.

public interface IAdresBeheerder

```
{
    bool BestaatAdres(Adres adres);

    bool BestaatGemeente(Gemeente gemeente);

    bool BestaatStraatnaam(string straatnaam, Gemeente gemeente);

    Adres SelecteerAdres(int id);

    List<Adres> SelecteerAdressenInGemeente(int NIScode);

    List<Adres> SelecteerAdressenInStraat(int straatID);

    Gemeente SelecteerGemeente(int NIScode);

    IList<Gemeente> SelecteerGemeenten();

    Straat SelecteerStraat(int id);

    List<Straat> SelecteerStratenInGemeente(int NIScode);

    List<Straat> SelecteerStratenInGemeente(string gemeentenaam);

    void UpdateAdres(Adres adres);

    void UpdateGemeente(Gemeente gemeente);

    void UpdateStraat(Straat straat);

    void VerwijderAdres(int id);

    void VerwijderGemeente(int NIScode);

    void VerwijderStraat(int id);

    void VoegAdresToe(Adres adres);

    void VoegGemeenteToe(Gemeente gemeente);

    void VoegStraatToe(Straat straat);
}
```