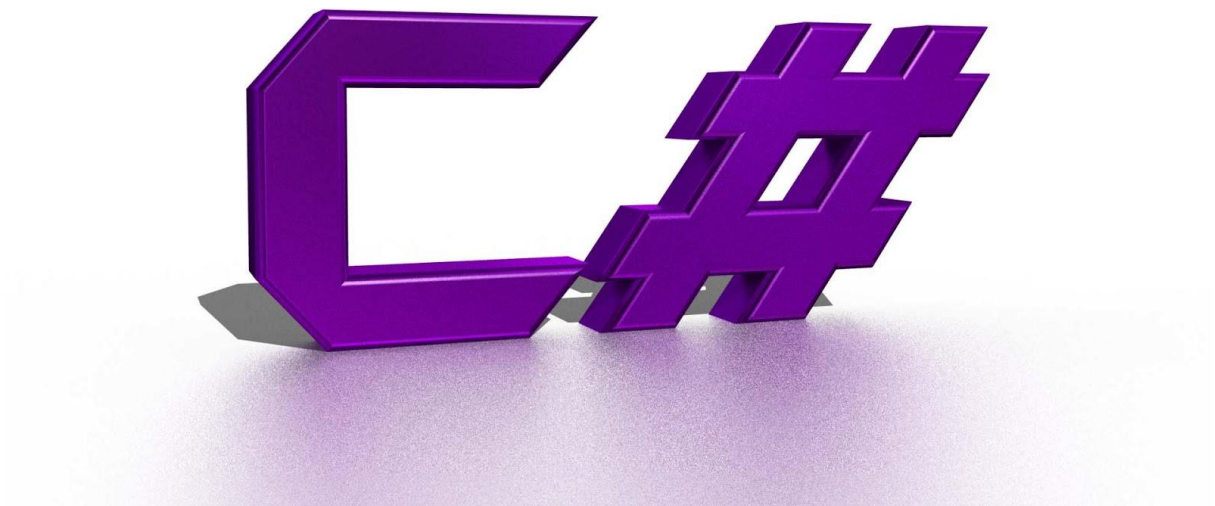


Ontwerpdocument



Naam	Sven Nottelman
Klas	S22
Datum	20-01-2016
Versie	4.0

Versiebeheer

Versienummer	Opmerkingen
1.0	Template
2.0	Invoegen inleiding
2.1	Invoegen ERD
2.2	Invoegen Strokendiagram en Databaseontwerp
3.0	Invoegen klassendiagram
4.0	Invoegen Implementatiemodel

Inhoudsopgave

1	Inleiding	3
2	Begrippenlijst.....	3
3	ERD	4
3.1	Diagram	4
3.2	Uitleg ERD	4
3.2.1	Project	4
3.2.2	Gebied	5
3.2.3	Bezoek	5
3.2.4	Waarneming	5
3.2.5	SoortWaarneming	5
3.2.6	Diersoort.....	5
3.2.7	Dier	5
4	Strokendiagram	6
5	Database ontwerp	7
6	Klassendiagram.....	8
6.1	Diagram	8
6.2	Uitleg klassendiagram	8
6.3	Beschrijving per klasse	9
6.3.1	Project	9
6.3.2	Gebied	9
6.3.3	Bezoek	9
6.3.4	Waarneming	9
6.3.5	SoortWaarneming	9
6.3.6	Dier	9
6.3.7	Vogel.....	9
7	Implementatiemodel.....	10

1 Inleiding

Het ontwerpdocument bevat de basisgegevens van de te ontwikkelen applicatie. Dit document bevat het ERD, Strokendiagram, klassendiagram en de architectuur van de applicatie.

Samen met het analysedocument is het met deze informatie mogelijk om de applicatie te ontwikkelen.

Bij het ERD staat er een korte uitleg van de tabellen en welke functies de database heeft. Hier staan wel de relaties benoemd, maar nog geen foreign keys.

In het strokendiagram komen daar nog de foreign keys bij.

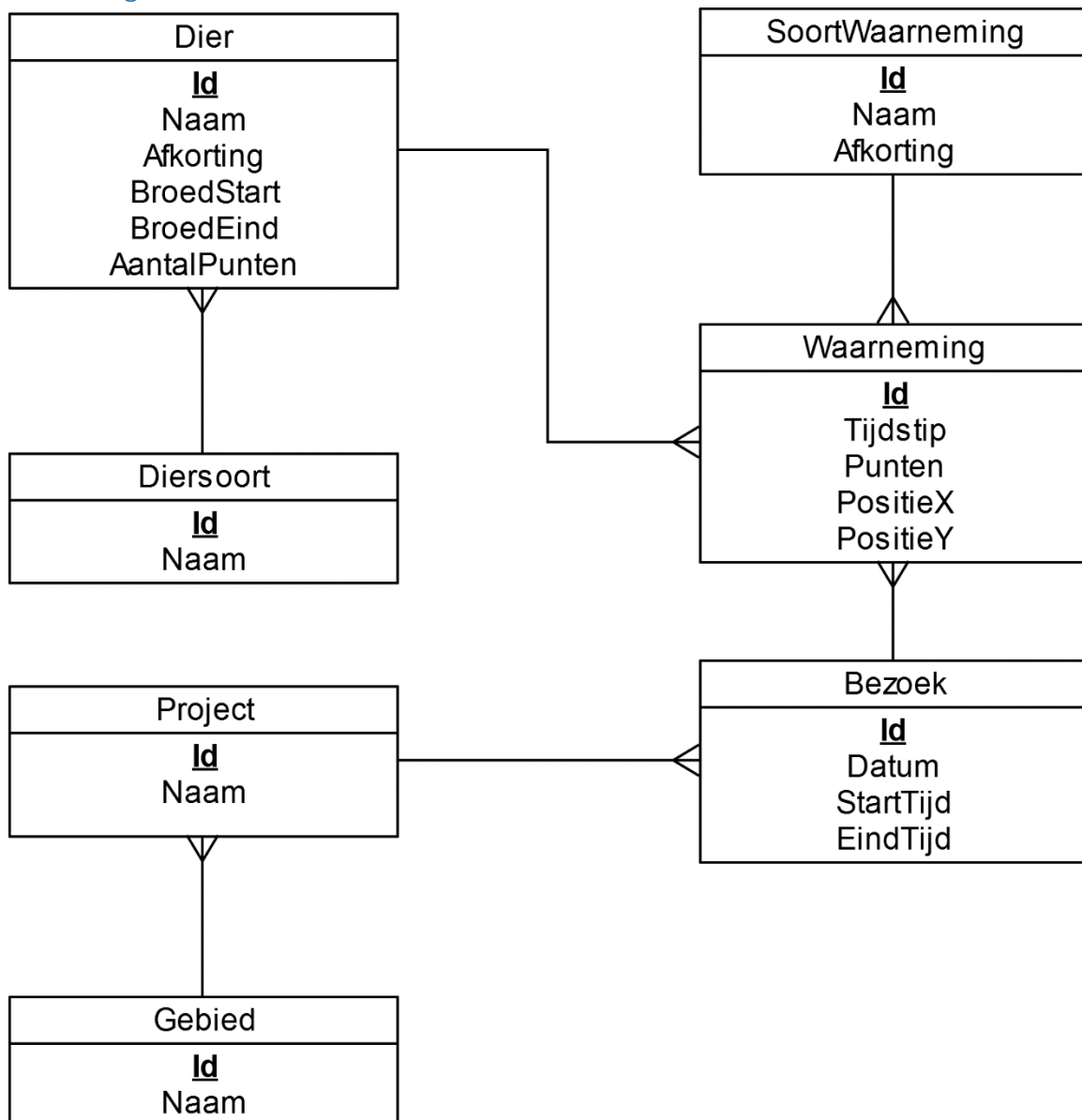
Het klassendiagram bevat alle informatie voor het systeem zelf. Hierin zijn alle klassen opgenomen en is het duidelijk hoe de applicatie in elkaar zit.

2 Begrippenlijst

- Entiteit
 - Een tabel in de database.
- ERD
 - Entity Relationship Diagram.

3 ERD

3.1 Diagram



3.2 Uitleg ERD

Het ERD bestaat uit 7 tabellen. Hieronder worden deze per tabel uitgelegd.

3.2.1 Project

De tabel Project heeft een ID en een Naam. Per project kunnen er meerdere bezoeken worden gedaan aan 1 gebied. Het veld ID is een number en het veld naam is een varchar2.

3.2.2 Gebied

Ieder gebied wat bestaat heeft een ID en een naam. Op een gebied kunnen er meerdere projecten zijn. Het veld ID is een number en het veld naam is een varchar2.

3.2.3 Bezoek

Een bezoek is een moment dat de vogelteller een bezoek brengt aan het gebied van zijn project. Van een bezoek worden de volgende zaken opgeslagen:

- | | | |
|-------------|-----------------------------|--------|
| - ID | uniek nummer | number |
| - Datum | over welke dag gaat het | date |
| - Starttijd | welke tijd is hij begonnen | date |
| - Eindtijd | welke tijd is hij geëindigd | date |

Ieder bezoek heeft meerdere waarnemingen, hiervoor is een aparte tabel.

3.2.4 Waarneming

Iedere waarneming is een soort waarneming, de mogelijkheden hiervan staan in een andere tabel opgeslagen. Verder wordt er van een waarneming de coördinaten onthouden (X en Y as). Dit is voor de plek op de kaart. Voor de waarneming zelf is er nog een puntenaantal en een tijdstip wat wordt opgeslagen.

- | | |
|------------|----------|
| - ID | number |
| - Tijdstip | date |
| - Punten | number |
| - PositieX | varchar2 |
| - PositieY | varchar2 |

3.2.5 SoortWaarneming

Deze tabel heeft de verschillende soorten van waarnemingen in zich zitten. Deze wordt gebruikt om in de tabel Waarneming aan te geven wat voor soort waarneming het is. Deze tabel heeft 2 velden. Een ID (number) en een naam (varchar2).

3.2.6 Diersoort

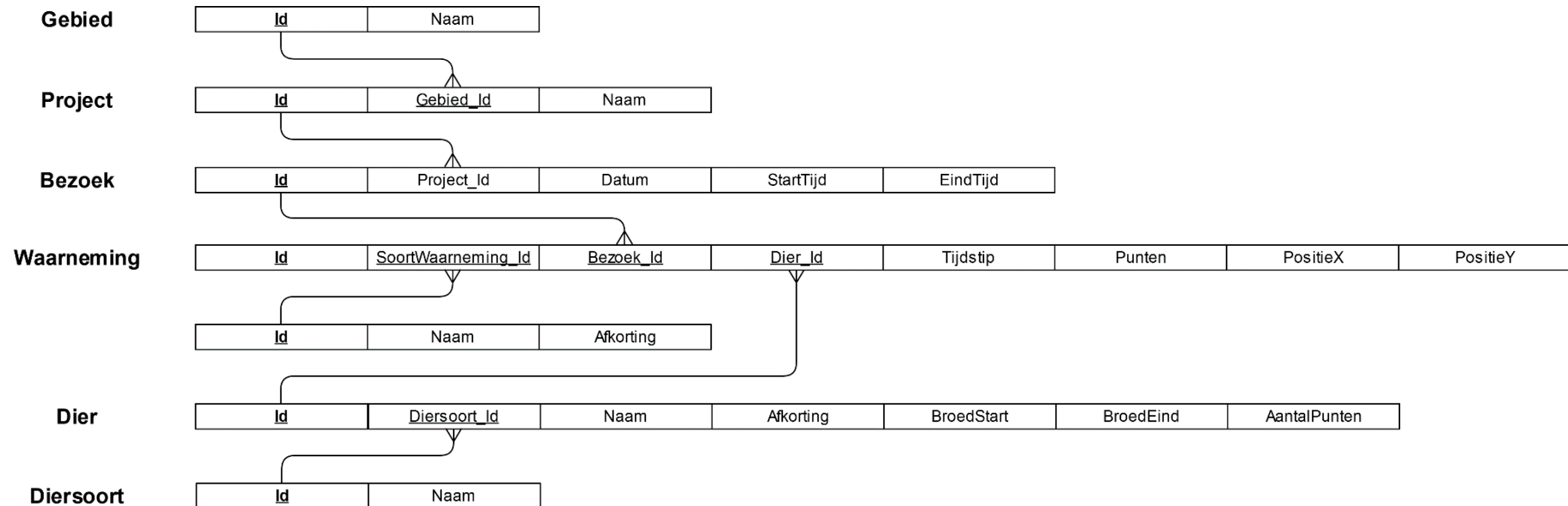
Een dier is in dit prototype altijd een vogel, maar de applicatie heeft de mogelijkheid dat er ook andere dieren in kunnen worden verwerkt, vandaar dat er een tabel diersoort is. Deze heeft een ID (number) en een naam (varchar2).

3.2.7 Dier

Naast de diersoort zoals in de tabel hierboven zijn er ook nog andere gegevens die worden opgeslagen van een dier. Dit gaat om een naam (varchar2), afkorting (varchar2), BroedStart en BroedEind datum (date) en het AantaPunten (number). De start en einddatum geven aan wanneer de broedperiode is en het aantal punten geeft aan hoeveel punten er nodig zijn voor 1 broedpaar.

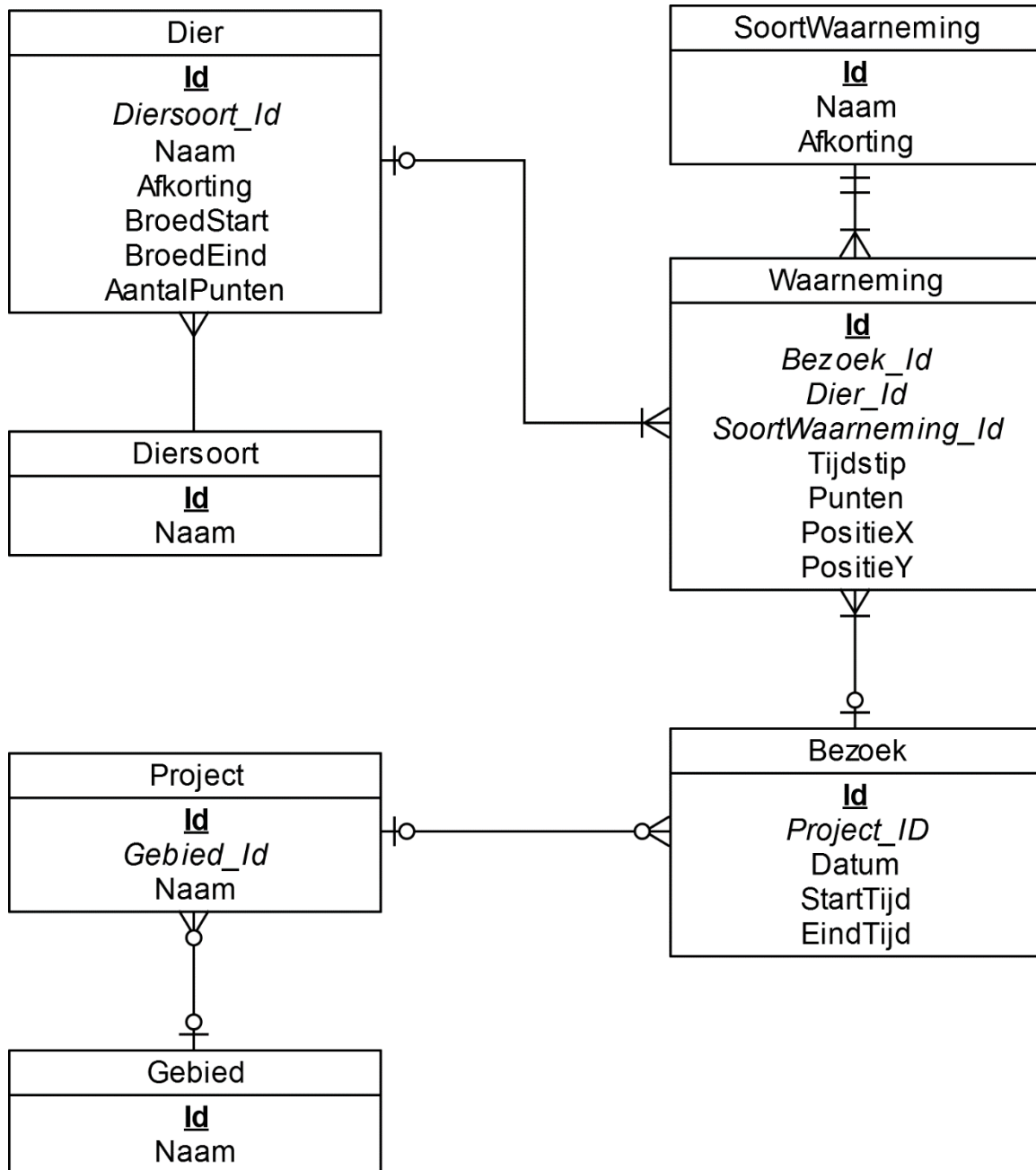
4 Strokindiagram

Het strokindiagram geeft aan waar alle foreign keys zitten en laat kort zien hoe de database in elkaar zit. Dit diagram heeft iets meer informatie dan het ERD. De uitleg van de tabellen staat bij het hoofdstuk ERD.



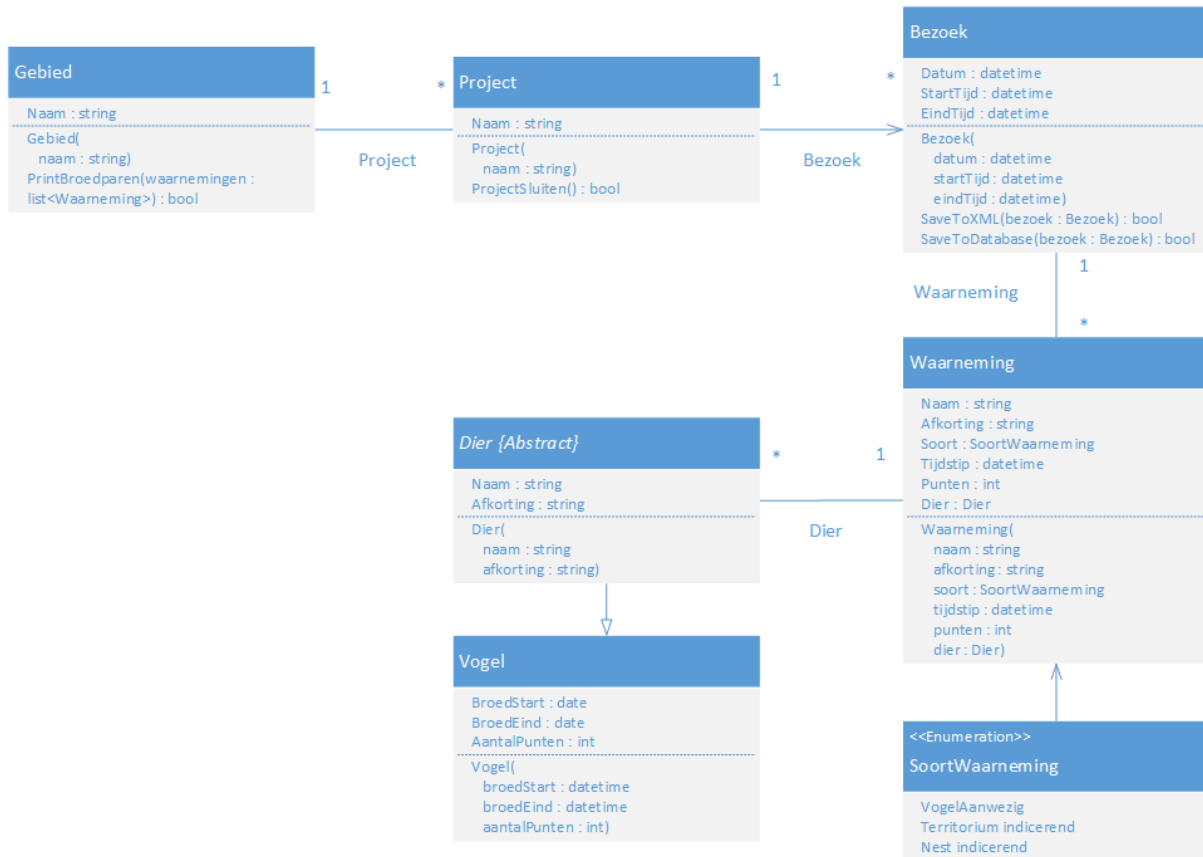
5 Database ontwerp

Bij het database ontwerp zijn de foreign keys toegevoegd, dit is hetzelfde als bij het strokendiagram. Het verschil hierbij is dat er hier ook minimale kardinaliteiten zijn toegevoegd. De uitleg van de tabellen staat bij het hoofdstuk ERD.



6 Klassendiagram

6.1 Diagram



Figuur 1 Eerste versie klassendiagram

Voor het prototype is dit de eerste versie van het klassendiagram. Dit zijn de models van de applicatie.

6.2 Uitleg klassendiagram

Ik heb de klassen gebaseerd op het databaseontwerp. Hierbij heeft iedere klasse zijn eigen rol. Deze worden gebruikt voor het opslaan en tijdelijk vasthouden van de gegevens die de gebruiker invoert.

Er wordt gestart met het maken van een project object. Ieder project heeft meerdere bezoeken. Deze staan opgeslagen in een lijst. Een bezoek kan weer meerdere waarnemingen hebben. Ook dit wordt op dezelfde manier opgeslagen. Bij een waarneming hoort een dier of meerdere dieren. Hier zit bij het prototype alleen een vogel. Maar later kunnen er meerdere dieren bijkomen. Vandaar dat er hier inheritance zit.

Op de volgende pagina staan alle klassen beschreven.

6.3 Beschrijving per klasse

6.3.1 Project

Een project heeft alleen een naam. Tevens worden er hier lijsten bewaard van bezoeken. In het prototype is er maar 1 project aanwezig. Dit project wordt als basis gebruikt.

6.3.2 Gebied

Een gebied heeft qua properties alleen een naam. Het gebied is altijd hetzelfde binnen een project. Ook heeft deze klasse een methode. Deze methode wordt gebruikt om het rapport te maken van het project. In het rapport staan alle broedparen van de vogels. Deze kunnen worden gesorteerd op Vogelnaam of op het aantal broedparen.

6.3.3 Bezoek

Een bezoek is een onderdeel van een project. Hierbij wordt de datum, start tijdstip en eind tijdstip opgeslagen. Tevens is er een lijst beschikbaar waarin alle waarnemingen in staan opgeslagen. Ook heeft deze klasse nog 2 methodes. Deze zijn voor het schrijven naar de XML file en naar het schrijven naar de database.

6.3.4 Waarneming

In een waarneming worden alle gegevens die belangrijk zijn voor de waarneming opgeslagen. Tevens wordt er een object van de klasse dier aangemaakt. Buiten het opslaan van de informatie over de waarneming wordt hier verder niks bijzonders gedaan in deze klasse.

6.3.5 SoortWaarneming

Iedere waarneming heeft ook een soort. Deze soort is een enum. Bij het maken van een object van de klasse waarneming wordt de soort gezet door middel van deze enum.

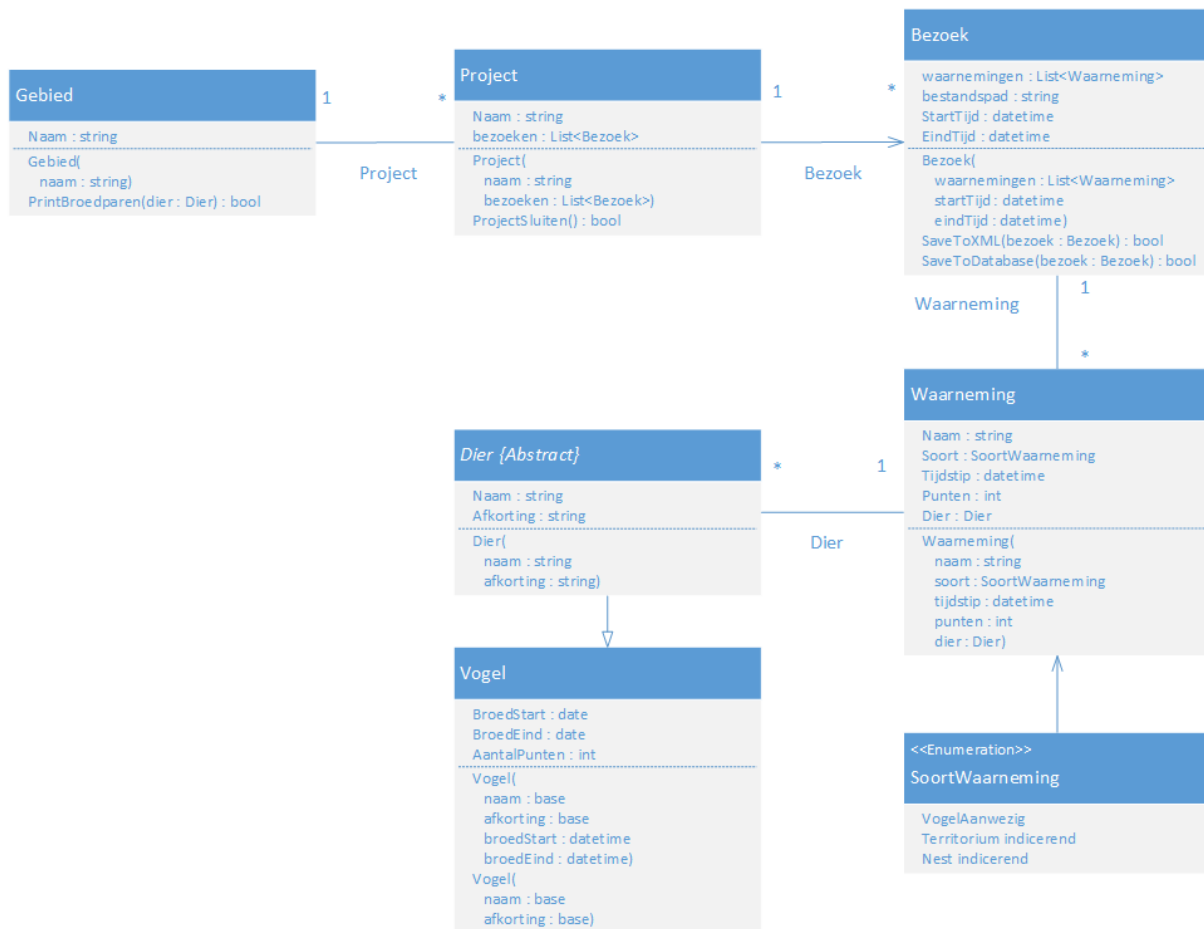
6.3.6 Dier

De klasse dier is abstract en slaat de basisgegevens op van een dier. Dit is de naam en de afkorting van een dier. Verder is er een subklasse waar de rest van de gegevens worden opgeslagen. In dit prototype wordt er alleen de subklasse vogel gebruikt. In de toekomst kunnen dit meer klassen zijn.

6.3.7 Vogel

De klasse vogel is een subklasse van de klasse dier. Hier wordt voor de klasse vogel de start en einddatum van de broedperiode aangegeven, tevens worden het aantal punten dat worden gegeven ingevuld.

7 Implementatiemodel



Figuur 2 Implementatiemodel (Gemaakt na het programma)

Dit is het implementatiemodel. Deze is gemaakt na het toevoegen van de functionaliteiten. In dit model zijn een aantal veranderingen die er vooral nog niet in zaten. Hieronder volgt een opsomming met de verschillen:

- Klasse: Gebied
 - o Methode: PrintBroedParen
 - De input is veranderd, nu gaat er een Dier object in I.P.V. een lijst met bezoeken.
- Project
 - o Field
 - Er is een lijst toegevoegd die de bezoeken bijhoudt binnen een project.
- Bezoek
 - o Field
 - Er is een lijst toegevoegd die de waarnemingen per bezoek bijhoudt.
- Waarneming
 - o Field
 - Het field "Afkorting" is verwijderd
- Vogel
 - o Constructor
 - Er is een tweede constructor toegevoegd met alleen een naam en een afkorting. Tevens zijn de velden die van de base afkomen ingevoegd.