

# POKEDEX

MICHELLE BROENS

## INHOUDSOPGAVE

Inleiding .....	3
Opdrachtschrijving .....	4
EER Model.....	5
Bijlage EER .....	6

## INLEIDING

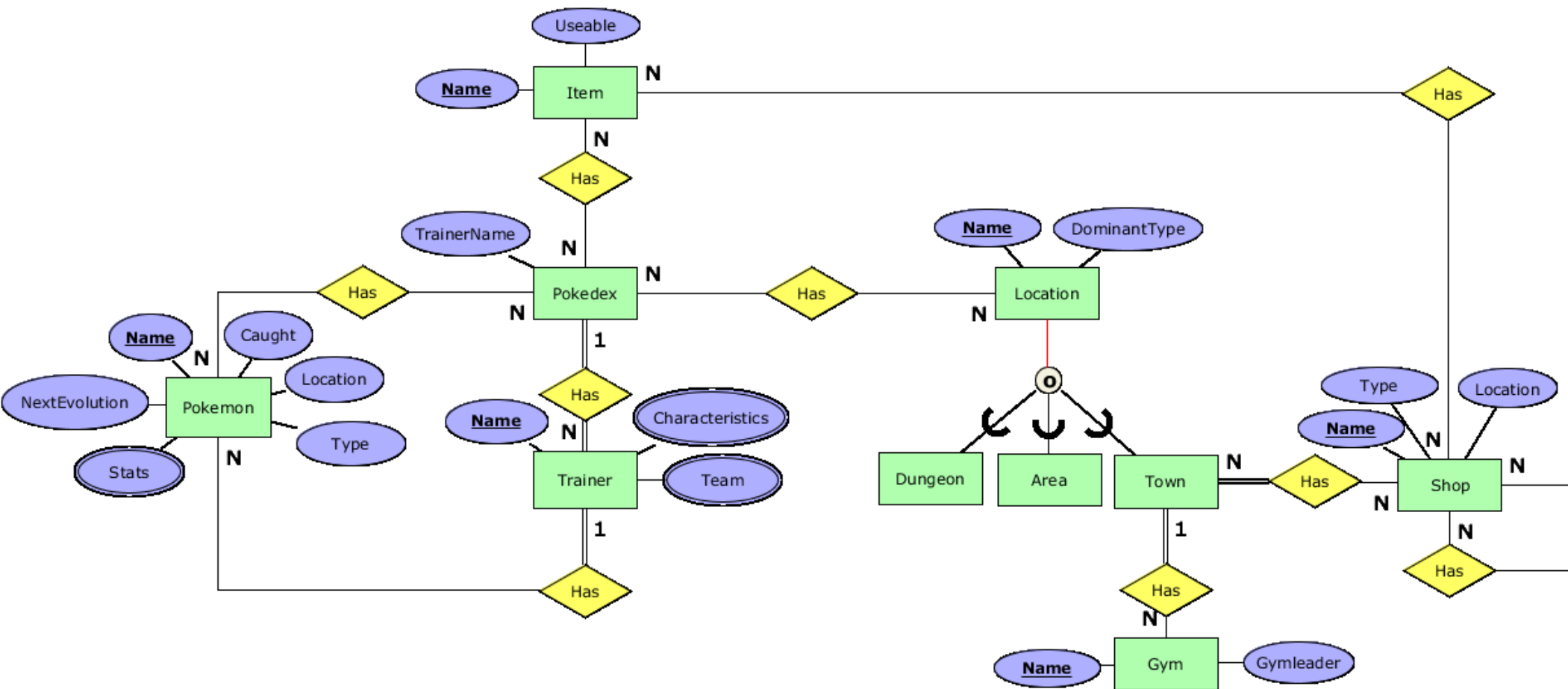
Voor deze nulmeting heb ik ervoor gekozen om een EER model te maken, omdat ik dit fijn vind om dit als eerste model te maken. Bij de andere modellen heb ik het gevoel dat je verder in het ontwikkelingsproces zit en al langer over een onderwerp nagedacht hebt. Ik vind een EER model heel erg ondersteunend bij het nadenken van hoe zit de applicatie die je gaat bouwen nu echt in elkaar, zo kun je snel zien hoe de basisstructuur van de applicatie in elkaar zit.

Ik heb het model gemaakt met de gedachte dat er niet maar één pokedex is, er zijn dus meerdere trainers die een verschillende pokedex hebben en niet zoals in de games, maar een pokedex voor je character.

## OPDRACHTOMSCHRIJVING

Een Pokédex is een apparaatje dat Pokémon-trainers bij zich hebben om informatie op te zoeken over de verschillende Pokémon, gyms en trainers. Ook kun je in de Pokédex zien welke Pokémon je al gevangen hebt of al gezien hebt. Indien de Pokémon gevangen is kun je zelfs nog zien op welke locatie je de Pokémon gevangen hebt. De Pokédex geeft onder andere per Pokémon aan van welke type hij is, welke stats hij heeft en in welke Pokémon hij kan evolueren. Daarnaast geeft de Pokédex ook aan op welke locatie een bepaalde Pokémon te vinden is. Een locatie kan een stad zijn, maar ook een straat of een gebied. Een stad kan tevens een Gym hebben, welke eigendom is van de gym leader. Daarnaast heeft de Pokédex een lijst van trainers met enkele eigenschappen van deze personen, denk aan naam, geslacht, badges enz. Ook kun je in de Pokédex informatie vinden over de verschillende voorwerpen die de trainer kan gebruiken, zoals bijvoorbeeld de Pokéflute, de Vishengel enz.

## EER MODEL



## BIJLAGE EER

De entiteit **Pokedex** is de koppeling tussen alle andere entiteiten, deze heeft de naam van de eigenaar erin staan zodat je kan zien van wie de pokedex is. In de pokedex staan items, deze hebben een **Name** en zijn **Useable**. Een usable Item is bijvoorbeeld een pokeball en niet useable item is een gymbadge. Een pokedex heeft meerdere trainers omdat je informatie van meerdere trainers moet kunnen bekijken, maar een trainer is de eigenaar van de pokedex. Je kan uiteraard ook informatie bekijken over verschillende pokemon. Er is ook informatie beschikbaar over verschillende locaties, dus gyms, stadjes en winkeltjes, dungeons en gebieden.

De pokedex heeft ook informatie over **Trainers**. Hiervan kun je kijken wat hun **Name** is, hun **Characteristics** (denk aan geslacht, geboortedatum enz.) en hun **Team**. Dit team is het team van **Pokémon** dat ze gebruiken.

Een trainer heeft meerdere **Pokémon** omdat hij meerdere gevangen kan hebben, de pokemon horen maar bij een trainer omdat maar een persoon de desbetreffende pokemon gevangen kan hebben. Een pokemon heeft een aantal eigenschappen. Hij heeft een **Name** ongeacht of de trainer hem een naam heeft gegeven of niet. Een **Caught** boolean om te kijken of de trainer deze pokemon al gevangen heeft, zo ja kun je zien op welke **Location** deze pokemon veel voorkomt. Je kunt ook zien wat voor **Type** de pokemon is, denk aan water, fire, dark enz. De **NextEvolution** is de naam van de pokemon waar de pokemon in evolueert, als dat er geen is blijft deze leeg.

Ik heb ervoor gekozen om een subtypering van **Location** te maken omdat je in **Dungeons** altijd Pokémon tegen kan komen en in **Towns** of **Area's** je per se in gras moet lopen om een Pokémon tegen te komen. Elke **Gym** komt maar een keer voor, maar niet in elke stad zit een gym. De **DominantType** bij locatie houdt in dat er in een gebied rond een vulkaan bijvoorbeeld bijna alleen maar fire type pokemon zitten.

Een **Shop** heeft een **Name** en een **Type**, deze type wil zeggen of het een item winkel is of bijvoorbeeld een winkel waar je pokeballs kan kopen. Ik heb ervoor gekozen om deze entiteit toe te voegen omdat ik vond dat deze best wel belangrijk was voor de items en de stad omdat dit de hoofdfunctie is voor een stad en je kan dan ook makkelijk kijken in welke stad een bepaalde winkel zit. Er zit een recursieve relatie bij winkel omdat er in een pokecenter bijvoorbeeld een item winkeltje zit.

Ik ben ervan uit gegaan dat er in elke **Town** in ieder geval een **Shop** heeft, dit is dan het Pokecenter, ook al zitten er verder geen winkeltjes, er moet altijd een pokecenter zijn.