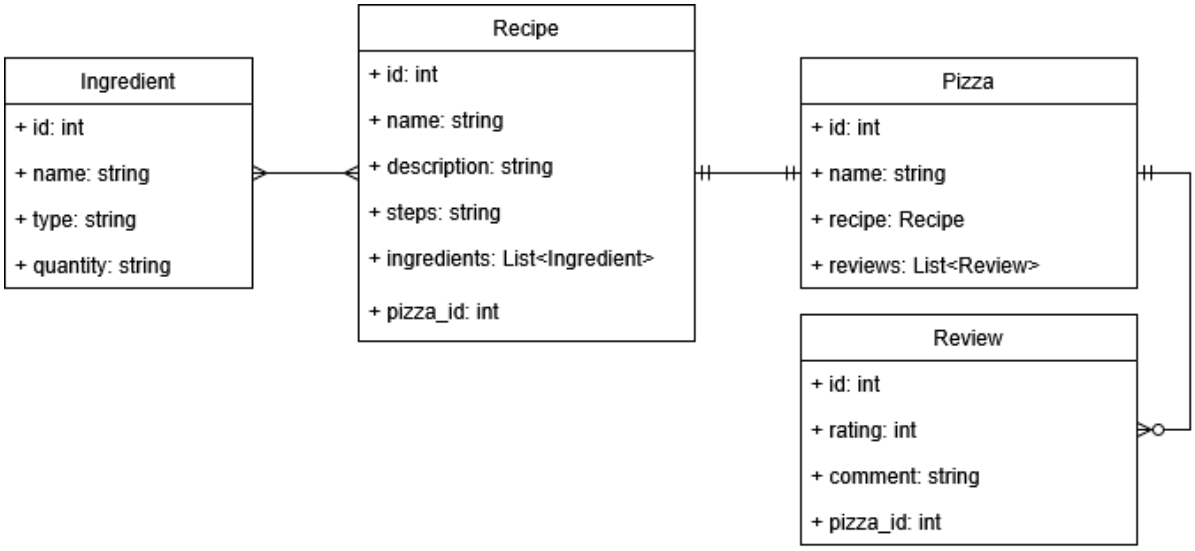
REST service workshop

Table of Contents

Class diagram



Er zijn 4 elementen te zien die allemaal wat met elkaar te maken hebben. Een pizza heeft een lijst aan reviews, en een recept. Elk recept heeft een of meerdere ingrediënten.

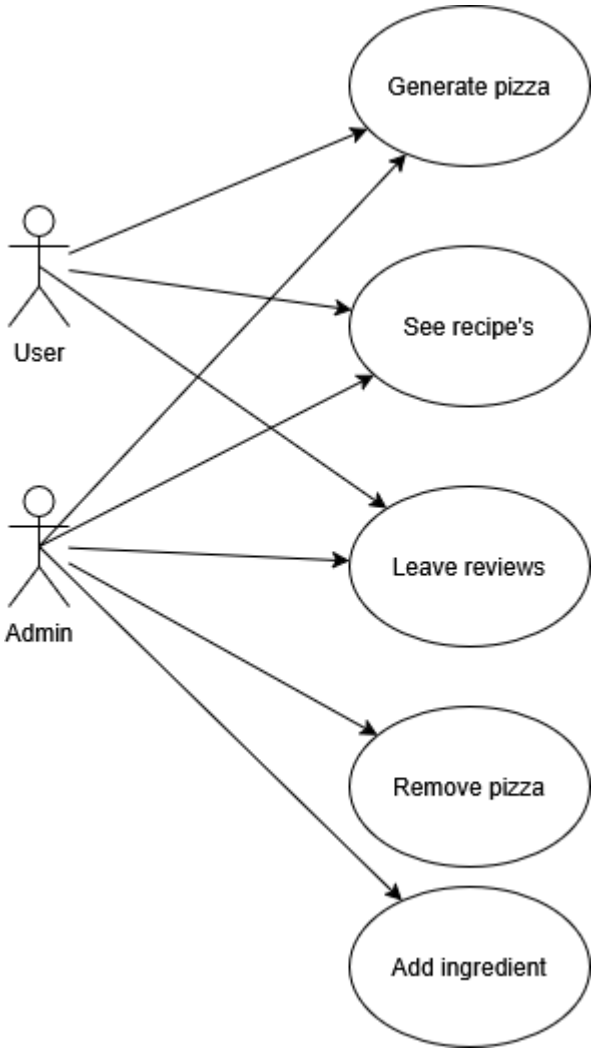
Een ingredient heeft 3 waardes. De naam van het ingredient, het type ervan, en de hoeveelheid die ervan gebruikt moet worden. De relatie met de recipe is een many to many, want een ingredient kan in meerdere recepten gebruikt worden. En een recept kan meerdere ingredienten hebben.

Het recept heeft 5 waardes. De naam van het recept, een omschrijving hiervan, de stappen benodigd om het te bereiden, de ingredienten die ervoor nodig zijn, en de bijbehorende pizza. Het recept heeft een one to one relatie met een pizza want een pizza kan maar 1 recept hebben en andersom ook.

De pizza heeft 3 waardes. De naam van de pizza, het recept dat erbij hoort, en de reviews die erbij horen. De pizza heeft een one to many relatie met reviews. Waarbij elke pizza meerdere reviews kan hebben, maar elke review om 1 pizza gaat.

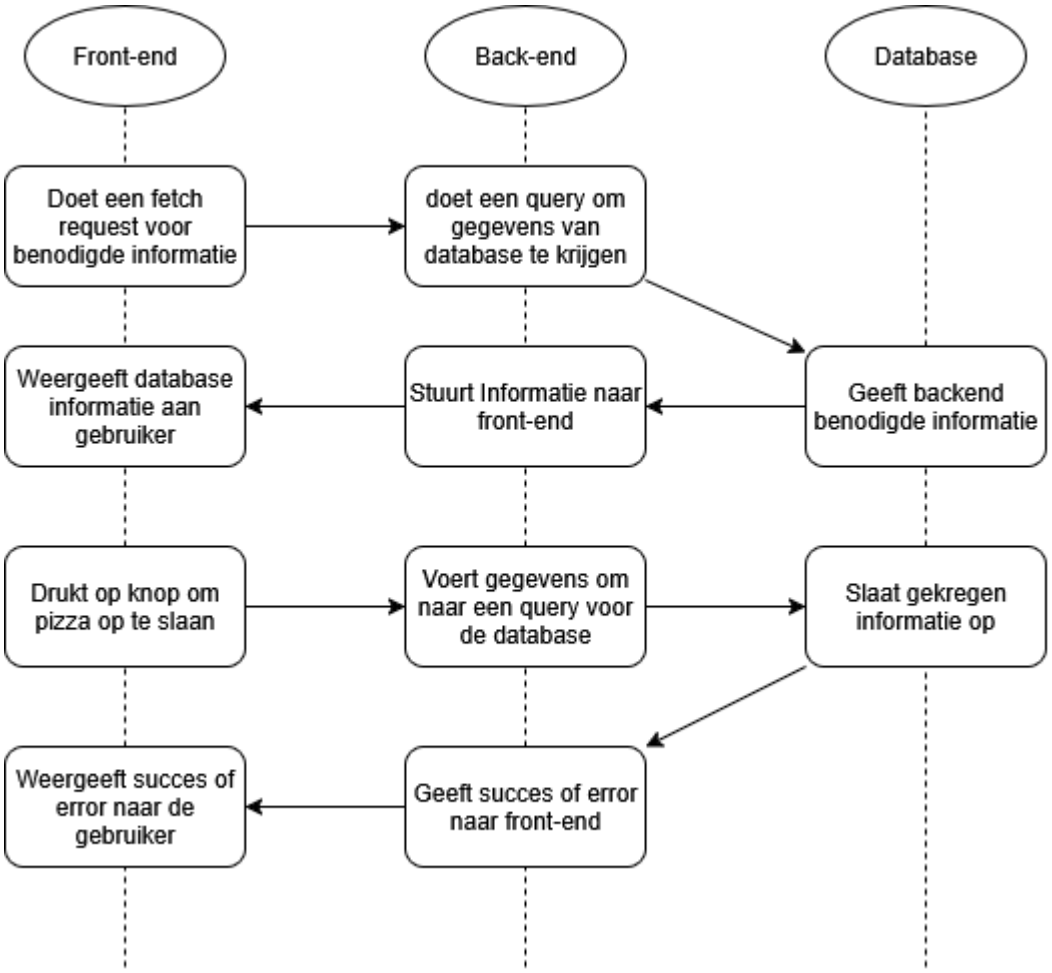
De review heeft 3 waardes. De gegeven rating, de comment over de pizza, en de bijbehorende pizza.

Use Case Diagram



Hierboven is een use case diagram te zien over het gebruiken van de web app. Er is te zien hoe iedereen een pizza kan genereren, een recept kan bekijken en een review achterlaten. Maar alleen een admin kan een pizza verwijderen of een ingredient toevoegen.

Sequence diagram



Hierboven is de sequence diagram te zien voor het inladen van infomratie vanuit de database, en het opslaan van een gegenereerde pizza.

Het weergeven van de informatie start bij de front end die een fetch request stuurt naar de gemaakte api in onze backend. Deze voert dan de juiste sql query uit naar de database, die deze informatie vervolgens terug geeft aan de backend. En deze verstuurd de info naar de front end.

Het opslaan van een gegenereerde pizza start bij het drukken op een knop in de front end. Deze knop doet een post request naar de backend voor de creatie van een pizza. De api zorgt ervoor dat de juiste query wordt gebruikt bij de database. En zo geeft de api door aan de front end of dit gelukt is of niet.

GET requests

GET	**/api/v1/pizzas**		
Returns a list of all pizza's			
Parameters:	**Name**	**Type**	**Description**
** required*	None	None	Returns a list of all pizza's
Responses:	**Code**	**Description / example if successful**	
	404	No pizza's found	
	200	List of all pizza's	

GET	**/api/v1/ pizzas / { id }**		
Return a specific pizza			
Parameters:	**Name**	**Type**	**Description**
** required*	id *	path	Id of the pizza to return
Responses:	**Code**	**Description / example if successful**	
	200	Details of the pizza	
	404	Pizza not found	

GET	/api/v1/recipes/{ id }/ingredients		
Returns a list of ingredients from a recipe by ID.			
Parameters:	**Name**	**Type**	**Description**
** required*	id	query	ID of the recipe
Responses:	**Code**	**Description / example if successful**	
	404	Recipe not found	
	200	List of ingredients of the recipe	

POST requests

POST	/api/v1/ pizzas		
Creates a pizza with a name and ID			
Parameters:		**Type**	**Description**
** required*			
Responses:	**Code**	**Description / example if successful**	
	200	Pizza created succesfully	
	201	Pizza created with empty values	
	404	Recipe not found	

POST	**/api/v1/ ingredients**		
Creates an ingredient			
Parameters:	**Name**	**Type**	**Description**
** required*	?	query	?
Responses:	**Code**	**Description / example if successful**	
	201	Ingredient created with missing values	
	200	Ingredient created	

POST	**/api/v1/** pizzas/{id}/reviews		
Creates a review about a specific pizza.			
Parameters:	**Name**	**Type**	**Description**
** required*	Id	path	Pizza id
Responses:	**Code**	**Description / example if successful**	
	200	Review added	
	201	Review added with missing values	
	404	Pizza not found	

PUT requests

PUT	**/api/v1/ pizzas/{id}**		
Replace a pizza by id			
Parameters:	**Name**	**Type**	**Description**
* *required*	Id	path	Pizza ID to replace
Responses:	**Code**	**Description / example if successful**	
	200	Pizza replaced	
	404	Pizza not found	

PUT	**/api/v1/ ingredients/{id}**		
Replace an ingredient by id			
Parameters:	**Name**	**Type**	**Description**
* *required*	Id	path	Ingredient to replace
Responses:	**Code**	**Description / example if successful**	
	200	Ingredient replace	
	404	Ingredient not found	

PUT	**/api/v1/ recipes/{id}**		
Replace a recipe by their id.			
Parameters:	**Name**	**Type**	**Description**
* *required*	Id	path	Id of recipe

Responses:	**Code**	**Description / example if successful**
	200	Recipe replaced
	404	Recipe not found

PATCH requests

PATCH	**/api/v1/ pizzas/{id}**		
Updates a pizza with the specified id			
Parameters:	**Name**	**Type**	**Description**
* *required*	?id	path	Id of pizza
Responses:	**Code**	**Description / example if successful**	
	204	Name updated	
	404	Pizza not found	

PATCH	**/api/v1/ reviews/{id}**		
Updates review by id			
Parameters:	**Name**	**Type**	**Description**
* *required*	?id	path	Id of review
Responses:	**Code**	**Description / example if successful**	
	204	Comment updated	
	404	Review not found	

PATCH	**/api/v1/ ingredients/{id}**
Updates an ingredient by a specific id	

Parameters:	**Name**	**Type**	**Description**
* *required*	id	path	Id of ingredient
Responses:	**Code**	**Description / example if successful**	
	200	?ingredient updated	
	404	Ingredient not found	

DELETE Requests

DELETE	**/api/v1/ pizzas/{id}**		
Deletes pizza by id			
Parameters:	**Name**	**Type**	**Description**
* *required*	id	path	Pizza id
Responses:	**Code**	**Description / example if successful**	
	204	Pizza deleted	
	404	Pizza not found	

DELETE	**/api/v1/ reviews/{id}**		
Deletes review by id			
Parameters:	**Name**	**Type**	**Description**
* *required*	id	path	Id of review
Responses:	**Code**	**Description / example if successful**	
	204	Review deleted	
	404	Review not found	

DELETE	**/api/v1/ ingredients/{id}**		
Deletes ingredient by id			
Parameters:	**Name**	**Type**	**Description**
* *required*	id	path	Id of ingredient
Responses:	**Code**	**Description / example if successful**	
	204	Ingredient deleted	
	404	Ingredient not found	