ontwerpdocument

SNIPPETGRAB



**Door:**

Max Meijer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Auteur | Wijziging |
| 1 | 18-11-2016 | M Meijer | Eerste documentversie |
| 2 | 7-1-2017 | M Meijer | Oplevering |
|  |  |  |  |

# Inhoud

Inhoud

[Inhoud 3](#_Toc467245775)

[Inleiding 4](#_Toc467245776)

[architectuur 5](#_Toc467245777)

[databaseontwerp 6](#_Toc467245778)

[Klassendiagram 7](#_Toc467245779)

[netwerk tekening 8](#_Toc467245780)

# Inleiding

In dit ontwerpdocument staat beschreven hoe de applicatie snippetgrab gemaakt zal worden. In dit document zullen duidelijke voorbeelden naar voren komen over hoe er te werk zal gaan, er zal een database-ontwerp in staan die een duidelijk beeld van onze database zal scheppen. In het analysedocument staat een EER die de basis zal vormen van ons database-ontwerp. Ook is er een klassendiagram waarin getoond word hoe de applicatie die gemaakt zal worden er in de basis uit zal zien. Als laatste staat er nog een netwerktekening die de omgeving laat zien waarop snippetgrab zal draaien.

# architectuur

Hieronder vindt u de architectuur van snippetgrab. Er wordt een duidelijke afscheiding gemaakt tussen de User interface, de logica van de applicatie en de DAL. De DAL zal in het repository pattern geschreven worden. Ook hiervan vindt u een UML voorbeeld uitgewerkt op de klasse User. Voor de andere klassen zal precies hetzelfde patroon worden gebruikt. De gehele applicatie zal ontwikkeld worden in ASP.NET.



# databaseontwerp

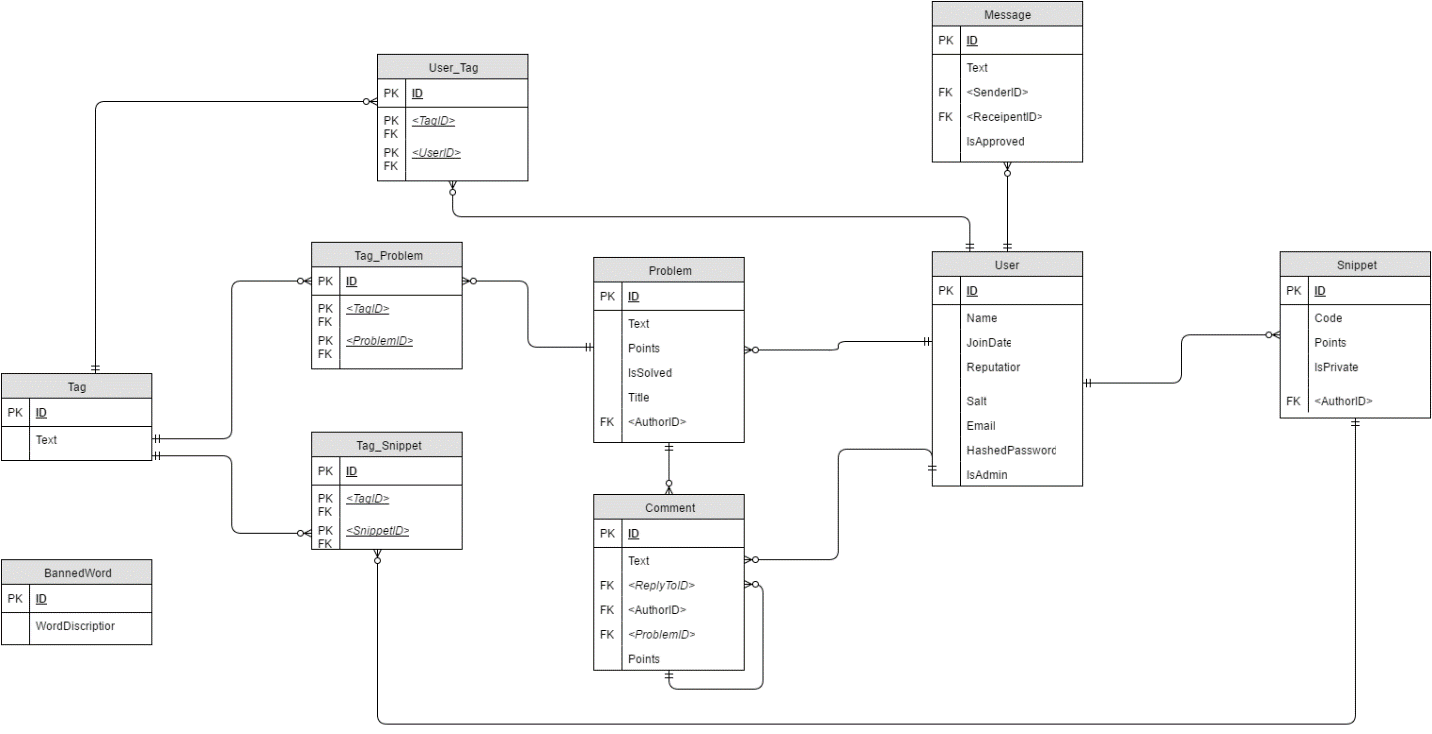
Hieronder vind u het database ontwerp voor snippetgrab, ik zal hier een korte toelichting bij geven. Het hoofd gedeelte van de applicatie bestaat uit gebruikers, snippets, problemen en comments.

Een gebruiker spreekt voor zich met een naam, wachtwoord, etc. het enige speciale is dat een gebruiker een reputation heeft dit krijgt hij doordat andere mensen hem punten toe kunnen dienen als hij bijvoorbeeld problemen oplost of goede bijdrages levert aan de gemeenschap. Zo kunnen mensen in één opslag zien of iemand betrouwbaar is of niet. Er zitten ook meerdere constraints op de user tabel. Het email adres moet Unique zijn en de reputatie mag nooit lager zijn dan 0.

Een snippet is een stuk code dat mensen kunnen uploaden om later weer te gebruiken. Ook deze bevat een punten aantal waarmee aangetoond kan worden of een stuk code goed is of niet.

Mensen die vragen hebben kunnen deze stellen d.m.v. een problem. Dit is gewoon een stuk tekst waarop andere mensen kunnen reageren om zo tot het juiste antwoord te komen. De recursieve lijn op de comment geeft aan dat een reactie op zichzelf kan worden gegeven oftewel een reactie. Stel dat iemand een antwoord wil verbeteren kan dat door erop te reageren.

De tags zijn voor de zoekfunctie. Mensen kunnen tags zoals “C#”, “C++”, etc. aan een probleem of snippet hangen waardoor deze later makkelijker te vinden is.



# Klassendiagram

Hieronder vindt u het klassendiagram van snippetgrab. Zoals u ziet komt het erg overeen met het databaseontwerp waar het ook grotendeels op gebaseerd is. Er zitten geen ingewikkelde dingen in dit klassendiagram dus veel toelichting zal niet nodig zijn.

Het punten aantal wat een object heeft word zoals in het analyse document beïnvloed door andere gebruikers.

C:\Users\Max\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Snippetgrab klassendiagram Oplevering.png

# netwerk tekening

De omgeving waar snippetgrab is opgezet bij het vak IN2 het bestaat uit 2 servers waarvan één functioneert als router naar de buitenwereld. Er zijn ook nog 2 clients aanwezig maar die laten we buiten beschouwing aangezien die niet van toepassing of invloed zijn op de servers.

De database en de .NET applicatie zullen beide lopen op de server “Harambe” welke u kunt vinden in het onderstaande netwerkschema.

