

# A.I.G.E.S

## Ontwerpdocument

**Door:** Gillvano Linakala

**Klas:** DB-S2

**Versie:** 0.3

## Inhoud

---

Inleiding .....	3
Versiebeheer .....	4
Architectuur .....	5
Database .....	6
ERD Diagram: .....	6
Database Diagram: .....	7
Klassendiagram .....	8

## Inleiding

---

Dit document bevat de architectuur, ERD Diagram en Klassendiagram voor het A.I.G.E.S project. Hier worden de interne werkingen van het applicatie beschreven qua database en functies.

Ik werk met de N-tier architectuur structuur zodat elke laag eigen verantwoordelijken heeft en daardoor makkelijker te testen is en ook dat het niet de hele applicatie breekt. Hierbij gebruik ik ook de S.O.L.I.D principes. Dependency inversion wordt in de architectuur hoofdstuk beshreven

## Versiebeheer

---

Versies van de applicatie en documentatie worden in [FreeGilio/AigesProject \(github.com\)](https://github.com/FreeGilio/AigesProject) aangepast en bijgehouden. Ook worden er versie updates in dit document toegevoegd om te zien wat er vanaf de vorige versies aangepast zijn.

Versie 0.3:

- Klassendiagram architectuur, SSMS ERD Diagram toegevoegd

Versie 0.2:

- Klassendiagram toegevoegd

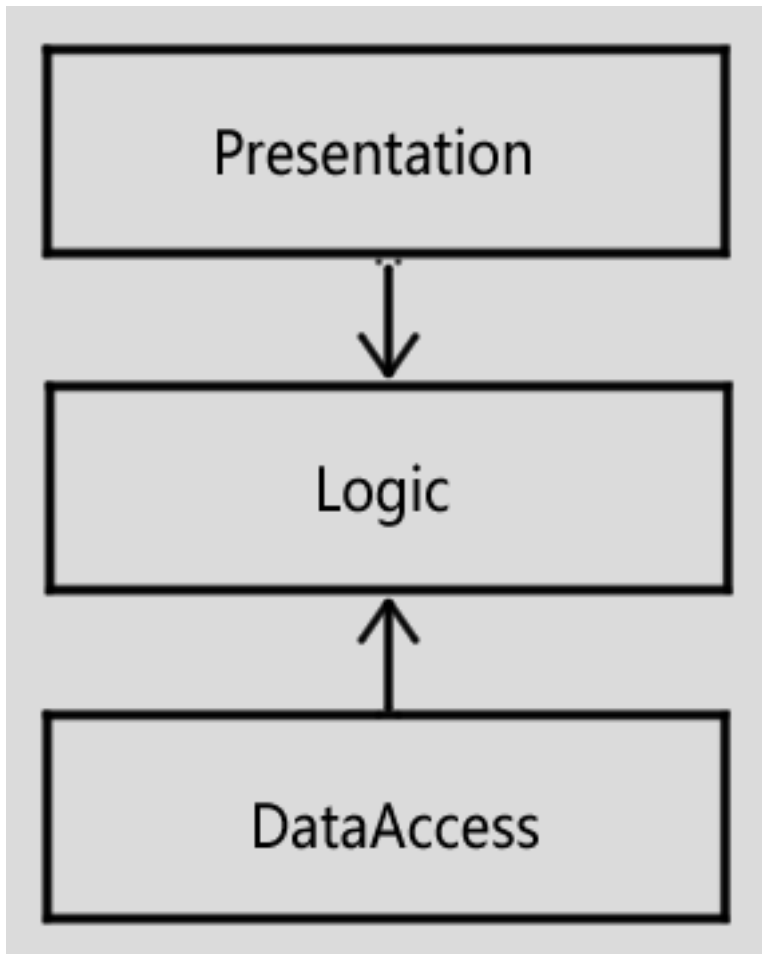
Versie 0.1:

- ERD Diagram toegevoegd

## Architectuur

---

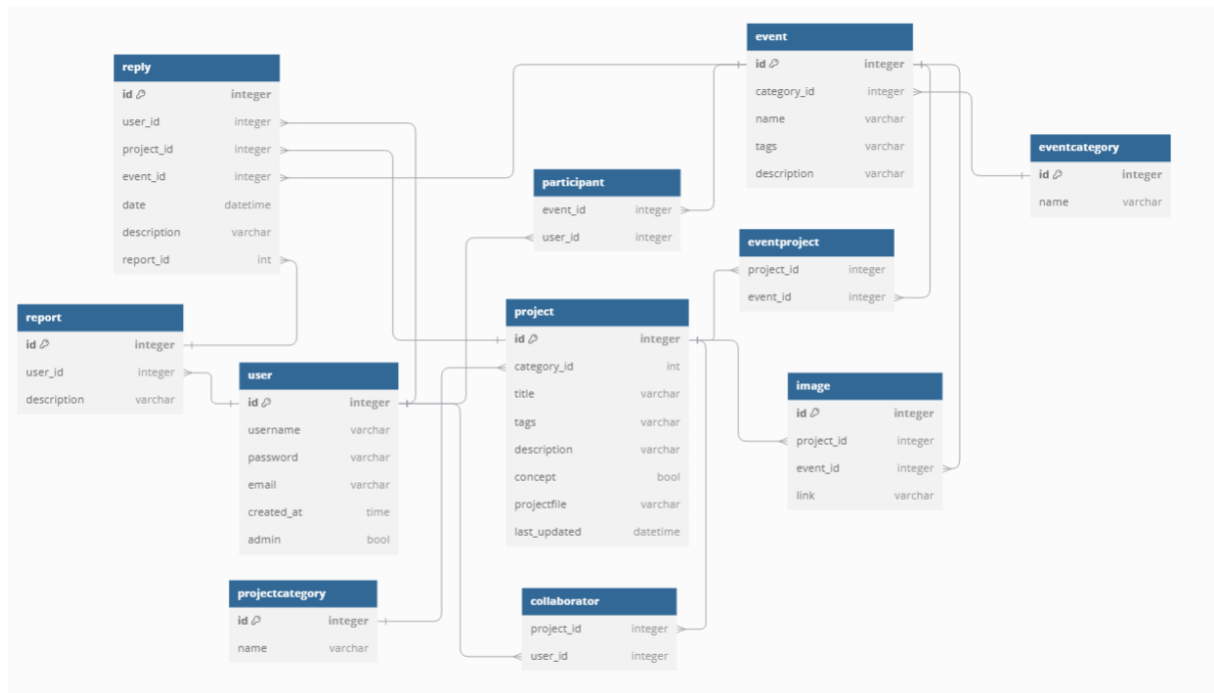
Voor mijn applicatie ga ik met MVC en het N-tier architectuur werken. Hierbij is er een Presentation, Logic en een DataAccess laag. Hierbij werk ik ook met dependency inversion. Dit betekent dat de logic laag niks van de andere lagen weet. Hiervoor worden er ook interfaces gebruikt in de Logic laag om zo gegevens van de DataAccess laag the pakken en logica uitvoeren.



# Database

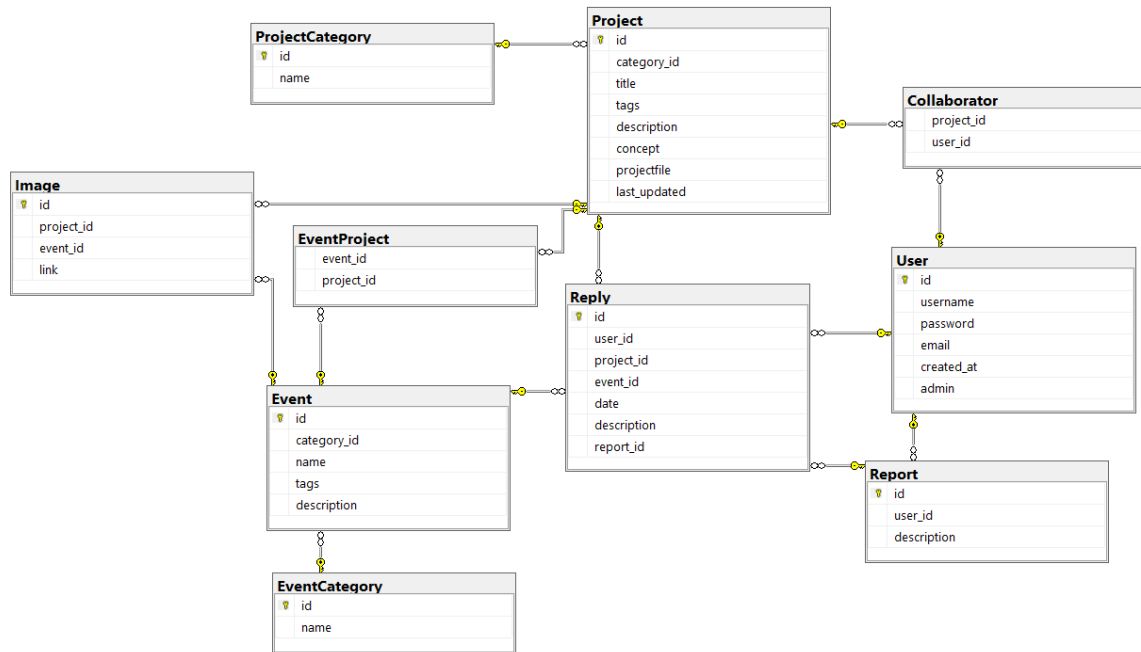
## ERD Diagram:

Het ERD Diagram is gebaseerd op de conceptueel model van mijn analyse document. Hier zijn er koppeltabellen gemaakt die allemaal een relatie hebben met Project en Events.



## Database Diagram:

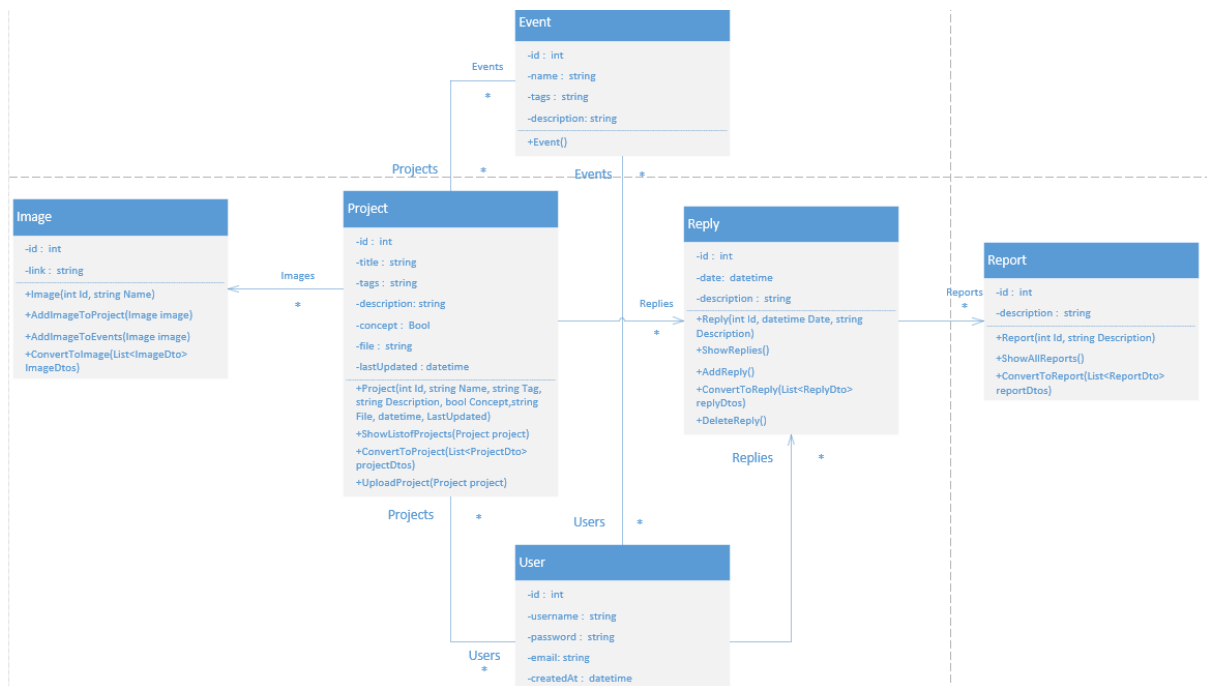
Het database diagram is hetzelfde als de ERD Diagram. De diagram hieronder is gemaakt in SQL Server Management Studio



# Klassendiagram

Hieronder staat twee klassen diagrammen van A.I.G.E.S die gemaakt is in Visio. Alle parameters hebben een public get en set.

Deze klassen diagram laat alleen de domeinmodellen en relaties ervan zien.





Dit is de voorlopige architectuur met "Project" als voorbeeld. Hier is ook al dependency inversion toegepast waarbij de Logic laag niks van de presentation en data lagen weet. Hier is in de applicatie ook dependency injection in de program.cs bestand toegepast.

Ik ga ook mijn domeinmodel met de viewmodel koppelen in de controller omdat ik in de domeinmodel validatie check logica wil toepassen. Bijvoorbeeld of het checkt of er geen titel ingevoerd is van een project.

