# Architecture

## ASP.NET MVC Architecture

Wij hebben voor dit project voor ASP.NET MVC gekozen simpelweg omdat dit een eis was van de opdrachtgever. Het is echter wel een verstandige keuze omdat ASP.NET makkelijk kan worden geïmplementeerd op het internet wat voor de eindgebruiker (de student/docent) het makkelijk maakt om ermee te werken zonder de benodigde software op zijn/haar pc te installeren.

MVC is een goede keuze aangezien we hierdoor makkelijk de views en de models uit elkaar halen waardoor aanpassingen in de view bijvoorbeeld makkelijk gedaan kunnen worden zonder alle models te hoeven wijzigen.

## Data acces Technology

Als Data acces hebben we de opdracht gekregen om Microsoft SQL Server 2012 te gebruiken. Dit is een verstandige keuze omdat we Linq kunnen gebruiken om code om te zetten naar SQL format en zo in het database kunnen stoppen, of om dingen op te halen.

## Samenwerking Views-Model

De views zijn allemaal verbonden met de controller en de controller is verbonden met de views. Als de model verandert, geeft hij dit door aan de controller en deze past dan automatisch de views aan zoals ze horen. Ook geldt dit andersom. Als de view verandert (bijvoorbeeld wanneer de user een ander tabblad opent), haalt de view m.b.v. de controller de goede data uit de models.

## Samenwerking Model-DB

De model worden gevuld met behulp van het database. Bijvoorbeeld als de eindgebruiker zijn cijfers wil zien, dan vragen de models de goede data (met behulp van Linq) uit het database. Deze wordt (zoals hierboven uitgelegd) doorgegeven aan de Views en zo krijgt de eindgebruiker de goede data op zijn scherm.