**<Naam Bedrijf opdrachtgever>**

**<Naam Project>**

**Project <nummer>**

**Technisch Ontwerp**

*In opdracht van* : <naam opdrachtgever>

*Auteur(s)* :

*Klas* :

*Datum* :

*Versienummer* :

*Plaats* :

Projectleider :

**Inhoud**

[1 Inleiding 4](#_Toc503265098)

[2 Klassendiagram 5](#_Toc503265099)

[3 Gestructureerde gegevensverzameling 6](#_Toc503265100)

[4 ERD 7](#_Toc503265101)

*<<<<<-------------------------- Deze bladzijden verwijderen ----------------------------->>>>>*

***Het Technisch Ontwerp***

*In het Technisch Ontwerp maak je de bouwtekening van het informatiesysteem dat je gaat realiseren. Je legt uit, aan de hand van het Functioneel Ontwerp (FO) en het Grafisch Ontwerp (GO) wat de gegevensstructuur van je applicatie wordt en welke technische oplossingen je kiest. Het is belangrijk dat je instructies zo duidelijk zijn dat iemand anders het informatiesysteem zonder problemen kan realiseren, zonder dat hij of zij vragen hoeft te stellen.*

*Een belangrijk onderdeel van het TO is de gegevensstructuur. Die leg je vast in een* ***klassendiagram*** *en in een* ***ERD****.*

*Daarnaast bepaal je de configuratie van de software. Daarmee worden versies en namen van gebruikte software bedoeld, maar ook instellingen, plug-ins, enz.*

*Werk tenslotte de configuratie van je systeem uit. Benoem bijvoorbeeld de keuzes voor software, de programmeertalen waar je gebruik van gaat maken, de keuze voor een specifiek CMS enz. Vermeld ook waarom je die keuzes hebt gemaakt. Vervolgens leg je uit hoe een collega ApplicatieOntwikkelaar het project moet gaan uitvoeren.*

*Houd er zoals bij elk verslag rekening mee, voor wie je het schrijft. Dit verslag schrijf je voor degene die het project gaat realiseren en voor de projectleider. Dat betekent dat de technische delen voldoende details bevatten, zodat degene die de volgende stap(pen) gaat uitvoeren hier alle benodigde informatie uit kan halen.*

*Taalgebruik: Zakelijk*

*Doelgroep: afhankelijk van het soort test, dus wel of geen vaktaal.*

*<<<<<-------------------------- Deze bladzijden verwijderen ----------------------------->>>>>*

# Inleiding

*De inleiding dient ervoor om aan de lezer kort uit te leggen waar dit project over gaat en wat de inhoud is van dit document.*

*De hele inleiding is maximaal één alinea (ca. 8 regels).*

Dit document is het Technisch Ontwerp voor het project Studentenregistratiesysteem ASTE Universiteit. In dit TO wordent allereerst het klassendiagram weergegeven dat is opgesteld aan de hand van het FO. Het hoofdstuk Genormaliseerde Gegevensverzameling laat de normalisaties per rapportage zien met aan het einde de samengevoegde normalisatie. Tenslotte kunt in in het hoofdstuk ERD het uitgewerkte EntiteitenRelatieDiagram vinden.

# Klassendiagram

*Een klassendiagram is een weergave van de objecten en hun relaties. Een object bestaat uit de volgende drie onderdelen.*

* *De naam van een object. Deze is uniek binnen het diagram.*
* *De attributen of eigenschappen van een object*
* *De operaties of methoden. De handelingen die verricht moeten worden op of door het object.*

*Het klassendiagram is een vertaling van de functionaliteiten uit het FO (requirements). Zorg er dus voor dat je ze allemaal verwerkt en niets vergeet!*

*Dit doe je als volgt: Stel aan de hand van de requirements een lijst op met kandidaat-klassen. Deze bekijk je vervolgens goed en je kijkt welke items wellicht attributen (eigenschappen) zijn van een ander item. Kijk ook naar super- en subklasses. Vervolgens kijk je welke attributen en welke methodes er nog meer bij de klassen horen en tot slot geef je de relaties tussen de klassen aan.*

*Voor een uitgebreide uitleg, zie het boek Applicatie- en mediaontwikkelaar van Hans van Rheenen, paragraaf 5.6 en 5.7.*

Hieronder volgt het klassendiagram van de ASTE Universiteit.



# Gestructureerde gegevensverzameling

*De volgende stap is dat je aan de hand van het klassendiagram en de requirements gaat kijken welke gegevens je in een tabel moet opslaan. Als vuistregel kun je ervan uitgaan dat elke klasse een mogelijke tabel is voor je database. Zodra je hebt bepaald welke tabellen er nodig zijn ga je invullen welke kolommen ze krijgen, ofwel welke eigenschappen er in worden opgeslagen. Dat doe je door allereerst de attributen uit de bijbehorende klasse over te nemen. Vervolgens neem je de requirements uit het FO nogmaals door en kijk je wat er nog meer in de tabellen moet om de requirements te kunnen uitvoeren.*

*Zie onderstaand voorbeeld. Hoewel je eigenlijk maar heel weinig info hebt, kun je met een beetje boerenverstand een heel eind komen.*

*De requirement is:*

* *Een student moet boeken kunnen aanvragen bij de bibliotheek die hem vervolgens worden toegestuurd.*

*Dan heb je een tabel STUDENT en een tabel BOEK nodig waar de student uit kan kiezen.*

*Van de STUDENT heb je info nodig om hem te identificeren, dus Studentnummer en volledige naam. Om hem het boek te kunnen sturen zijn de adresgegevens nodig. Dus adres, postcode en woonplaats. Tenslotte wil je hem ook nog kunnen benaderen als er wat mis is met de aanvraag. Daarvoor is een telefoonnummer en/of een emailadres nodig.*

*Van het BOEK heb je een uniek nummer (per exemplaar) nodig voor de administratie: Het uitleennummer. Om te weten welk boek (titel) het is, is de titel en de auteur nodig. Om het zeker te weten leg je ook het ISBN vast. Het is gebruikelijk om bij boektitels ook de uitgever en het jaar van uitgifte te vermelden, maar die gegevens komen nergens in de requirement voor, dus die laten we weg.*

*Neem alleen die velden op die nodig zijn. Bedenk dus niet zelf attributen of velden.*

Uit de functionaliteiten uit het FO en uit het klassendiagram volgen de volgende tabellen.

**STUDENT BOEK**

Studentnummer Uitleennummer

Voornaam ISBN

Tussenvoegsel Titel

Achternaam Auteur

Adres

Postcode

Plaats

Telefoonnummer

Email

# ERD

*Als je alle tabellen hebt, geef je de relaties aan tussen die tabellen.*

*Het EntiteitenRelatieDiagram is de grafische weergave van je data. Het bestaat uit twee onderdelen.*

* *Entiteiten (tabellen): een entiteit heeft altijd een naam en eigenschappen (attributen), overgenomen uit de normalisatie.*
* *Relaties tussen entiteiten: deze relaties geven weer welke relatie er tussen twee entiteiten is. Een relatie is altijd een-op-een, een-op-meer of meer-op-meer.*

*Neem de entiteiten uit het vorige hoofdstuk over en geef de relaties tussen die entiteiten aan.*

*Vervolgens vul je dit ERD aan met de gegevens uit het klassendiagram.*

Het ERD voor de ASTE Universiteit ziet er als volgt uit:

