



I-Talent platform

Analyse

Departement PXL-IT

www.pxl.be

Team 1

Arjen Schuurmans

Bart Hunerbein

Dennie Grondelaers

Jesse Vranken

Niek Vandael

Academiejaar 2015-2016

Inleiding

Hogeschool PXL zou graag gebruik maken van een online systeem om het i-talent project voor zowel studenten als docenten makkelijker te maken.

Studenten van verschillende departementen van hogeschool PXL moeten een verplicht aantal uren werken aan een zelf gekozen project. Projecten kunnen niet altijd slechts door 1 of enkele studenten verwezenlijkt worden, daarom zou een online platform waarop studenten ideeën kunnen lanceren en/of zich hierop inschrijven uitkomst kunnen bieden. Samenwerking met verschillende departementen kan op die manier bevorderd worden en studenten hebben de mogelijkheid om deel te nemen aan projecten die in de lijn van hun interesses liggen.

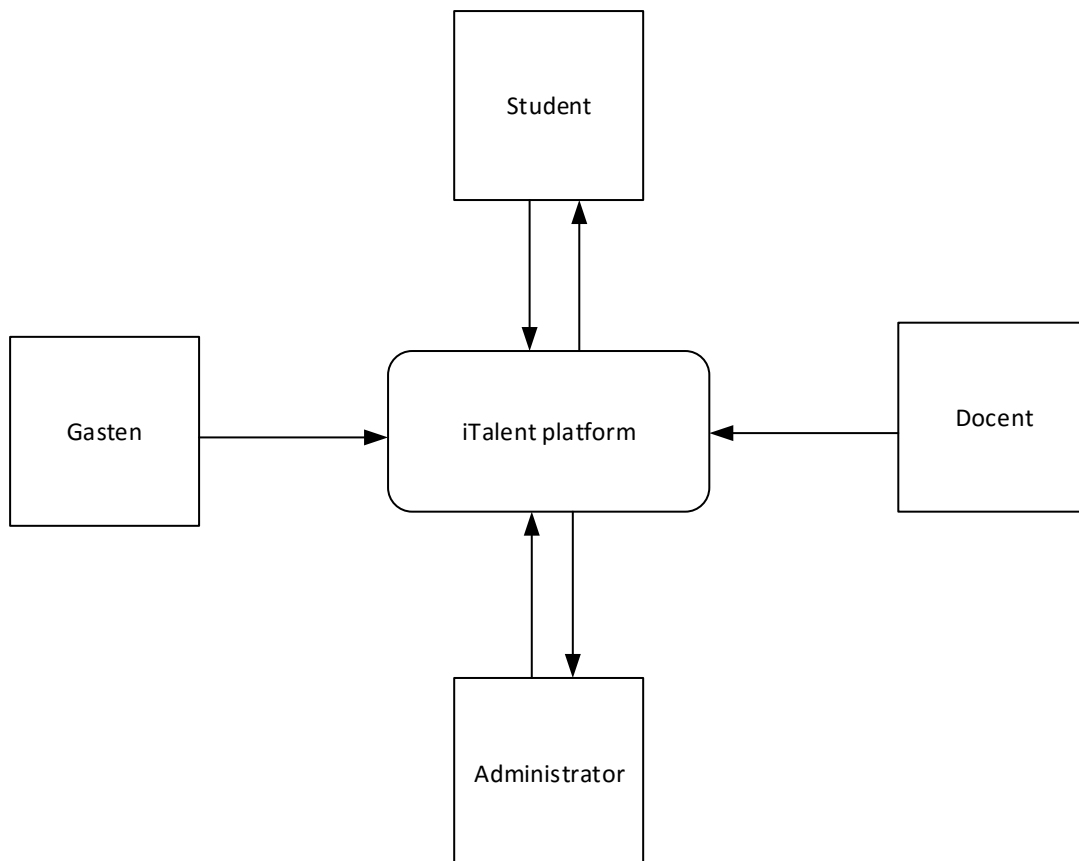
Uiteraard is niet de bedoeling dat ieder idee zomaar gelanceerd kan worden. Hiervoor is allereerst toestemming nodig van begeleidende docenten. Docenten kunnen projecten goedkeuren, monitoren en eventueel ondersteunen.

Als u verder kijkt zal u een inhoudstabel zien met daarop een duidelijke weergave op welke pagina's we met welke analyse gewerkt hebben.

Inhoudstafel

Inleiding	3
Hoofdstuk 1 - DFD	5
1.1 Contextdiagram.....	5
1.2 Gebeurtenissenlijst.....	6
1.2.1. Administrator definieert categorieen.....	7
1.2.2. Administrator definieert gebruikers.....	7
1.2.3. Administrator definieert rollen.....	7
1.2.4. Administratoren kunnen gebruikers verwijderen.....	7
1.2.5. Docenten, studenten en gasten kunnen een projectenlijst raadplegen.....	8
1.2.6. Docenten, studenten en gasten kunnende projectenlijst rangschikken.....	9
1.2.7. Docenten, studenten en gasten kunnen projectdetails opvragen.....	10
1.2.8. Studenten, docenten en administratoren authenticeren in de applicatie.....	11
1.2.9. Docenten, studenten en gasten kunnen lijsten opvragen van lopende projecten.....	12
1.2.10. Docenten, studenten en gasten kunnen lijsten opvragen van afgeronde projecten.....	13
1.2.11. Docenten en studenten kunnen kunnen projecten liken.....	14
1.2.12. Studenten kunnen nieuwe projecten aanmaken.....	15
1.2.13. Studenten en docenten kunnen inschrijven op projecten.....	16
1.2.14. Studenten kunnen hun eigen projecten bewerken.....	17
1.2.15. Studenten kunnen hun eigen projecten verwijderen.....	17
1.2.16. Docenten kunnen projecten goedkeuren.....	18
1.2.17. Docenten kunnen projecten publiek maken.....	18
1.2.18. Studenten en docenten kunnen de status van projecten volgen.....	19
1.3 Gegroepeerde gebeurtenissen.....	20
1.3.A. Beheer iTalent gegevens.....	21
1.3.B. Beheer iTalent authenticatie.....	22
1.3.C. Beheer projectenlijst.....	23
1.3.D. Beheer projectdetails.....	24
1.3.E. Beheer projecttoegang.....	25
Hoofdstuk 2 – Normalisatie	27
2.1 Normalisatie.....	27
2.2 Commentaar Normaalvorm 1 & 2.....	28
2.2 Commentaar Normaalvorm 3.....	29
Hoofdstuk 3 – ERD	30
3.1 ERD.....	30
3.2 Commentaar ERD relatie 1-7.....	31
3.2 Commentaar ERD relatie 8, 9.....	
Hoofdstuk 4 – Logisch gegevensmodel	
4.1 Logisch gegevensmodel.....	
4.1 Commentaar logisch gegevensmodel.....	

Contextdiagram

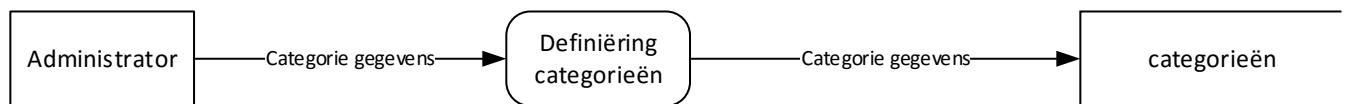


De actoren die inwerken op het iTalent systeem zijn de docenten en studenten van de pxl. Verder kunnen ook gasten beperkte acties ondernemen. Ten slotte zijn er de Administrators die enkele unieke acties kunnen uitvoeren.

Gebeurtenissenlijst

1. Administrator definieert categorieën
2. Administrator definieert gebruikers
3. Administrator definieert rollen
4. Administratoren kunnen gebruikers verwijderen
5. Docenten, studenten en gasten kunnen een projectenlijst raadplegen
6. Docenten, studenten en gasten kunnende projectenlijst rangschikken
7. Docenten, studenten en gasten kunnen projectdetails opvragen
8. Studenten, docenten en administratoren authenticeren in de applicatie
9. Docenten, studenten en gasten kunnen lijsten opvragen van lopende projecten
10. Docenten, studenten en gasten kunnen lijsten opvragen van afgeronde projecten
11. Docenten en studenten kunnen kunnen projecten liken
12. Studenten kunnen nieuwe projecten aanmaken
13. Studenten en docenten kunnen inschrijven op projecten
14. Studenten kunnen hun eigen projecten bewerken
15. Studenten kunnen hun eigen projecten verwijderen
16. Docenten kunnen projecten goedkeuren
17. Docenten kunnen projecten publiek maken
18. Studenten en docenten kunnen de status van projecten volgen

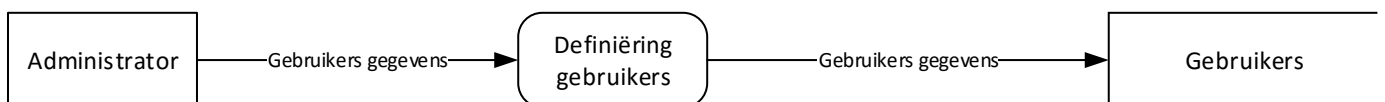
1. Administrator definieert categorieën



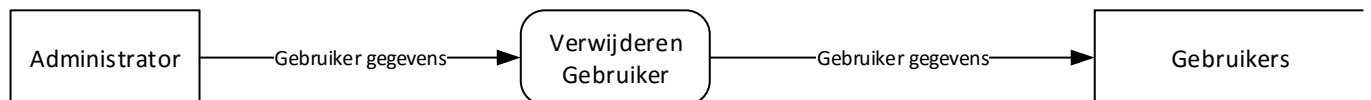
2. Administrator definieert rollen



3. Administrator definieert gebruikers

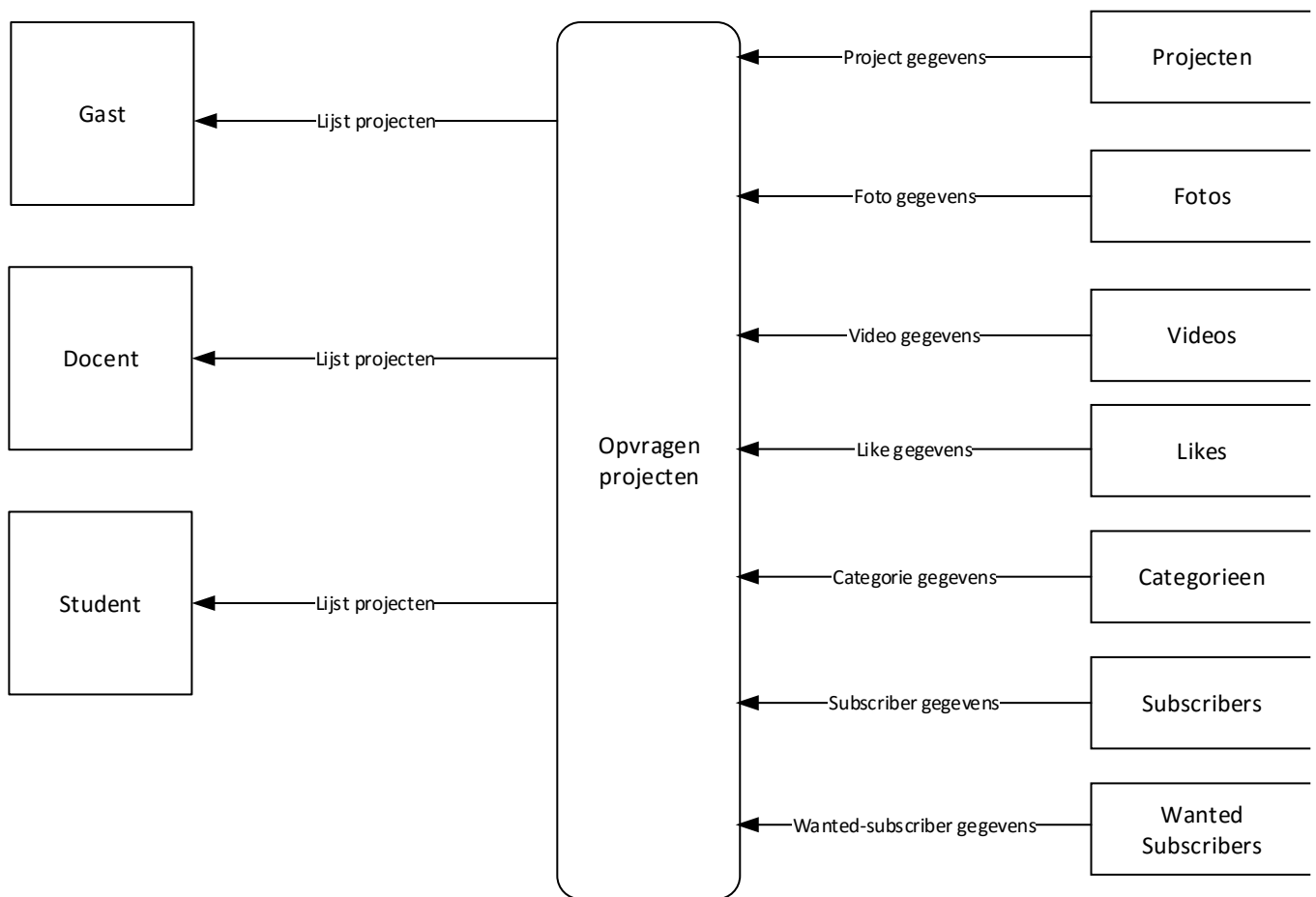


4. Administratoren kunnen gebruikers verwijderen



Vooralleer het iTalent platform van start kan gaan heeft het systeem informatie nodig. De administatoren, hoofd van iTalent, kan via deze procedure makkelijk de stamgegevens aanpassen. Indien er wijzigingen gebeuren kan de administator via deze data stores alles makkelijk aanpassen.

5. Docenten, studenten en gasten kunnen een projectenlijst raadplegen

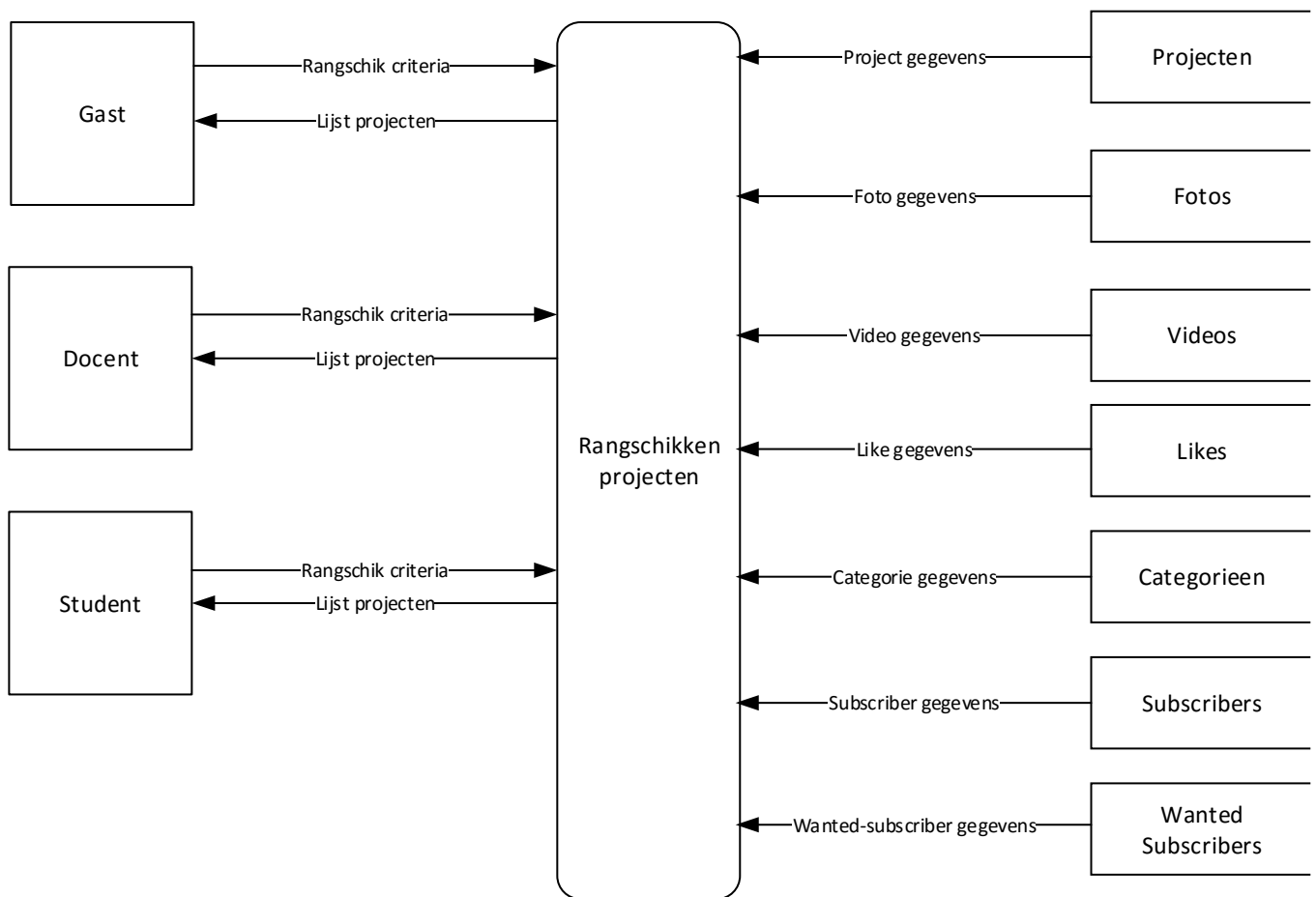


Docenten, studenten en gasten kunnen een projectenlijst raadplegen.

Gasten kunnen enkel de publieke projecten bekijken.

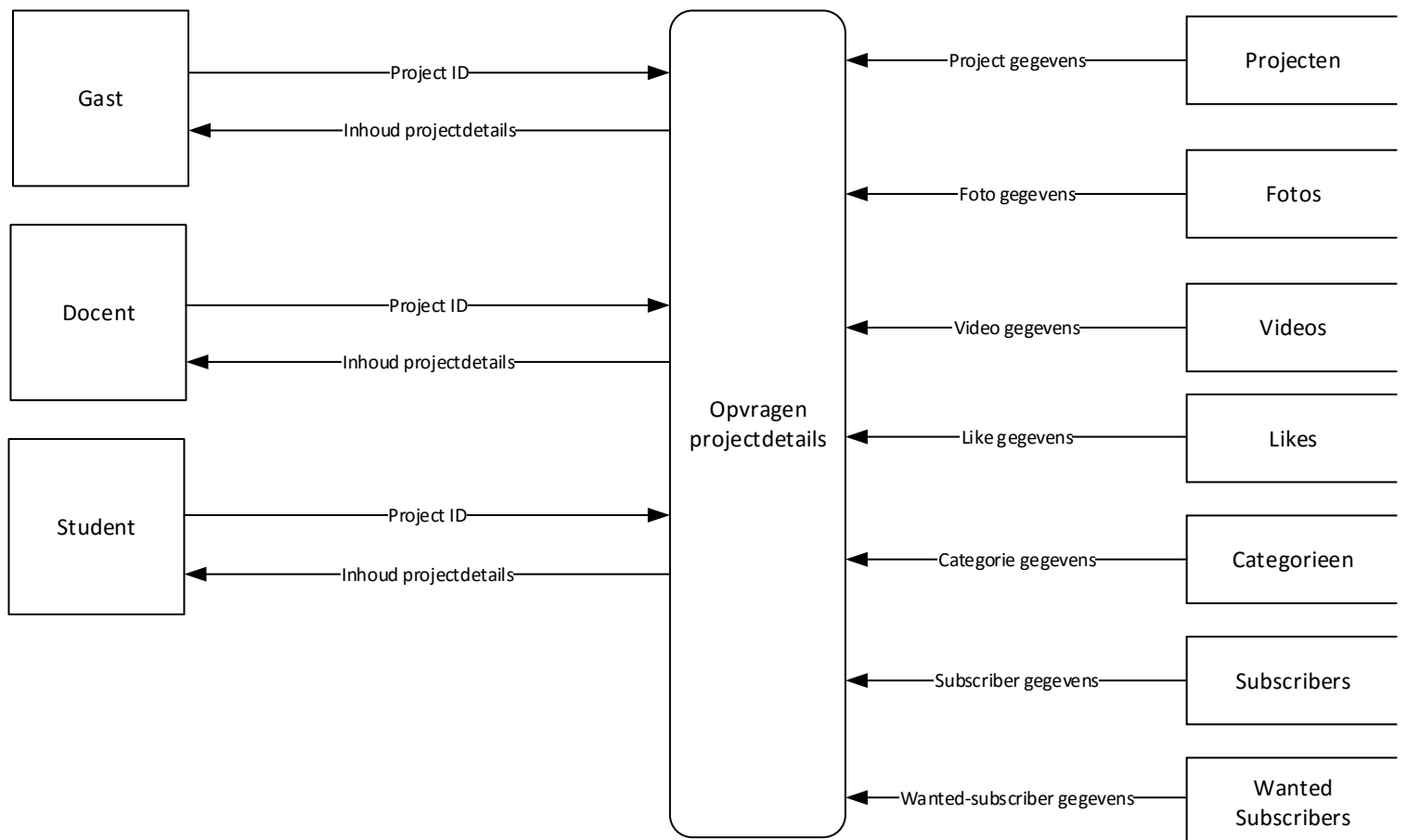
Studenten kunnen enkel geverifieerde projecten bekijken die zowel publiek als private status hebben.

6. Docenten, studenten en gasten kunnende projectenlijst rangschikken



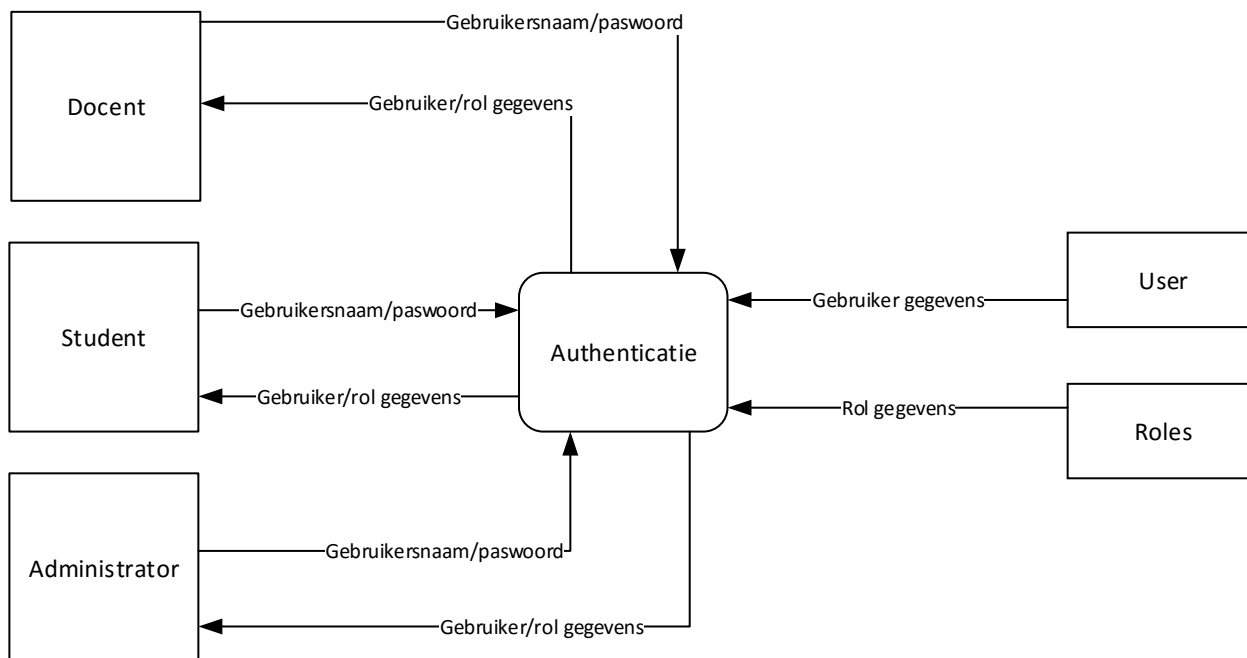
Docenten, studenten en gasten hebben de mogelijkheid om via door hun gespecificeerde zoekparameters projecten te rangschikken. Zo kunnen bijvoorbeeld projecten gerangschikt worden op aantal likes of op een specifieke categorie.

7. Docenten, studenten en gasten kunnen projectdetails opvragen



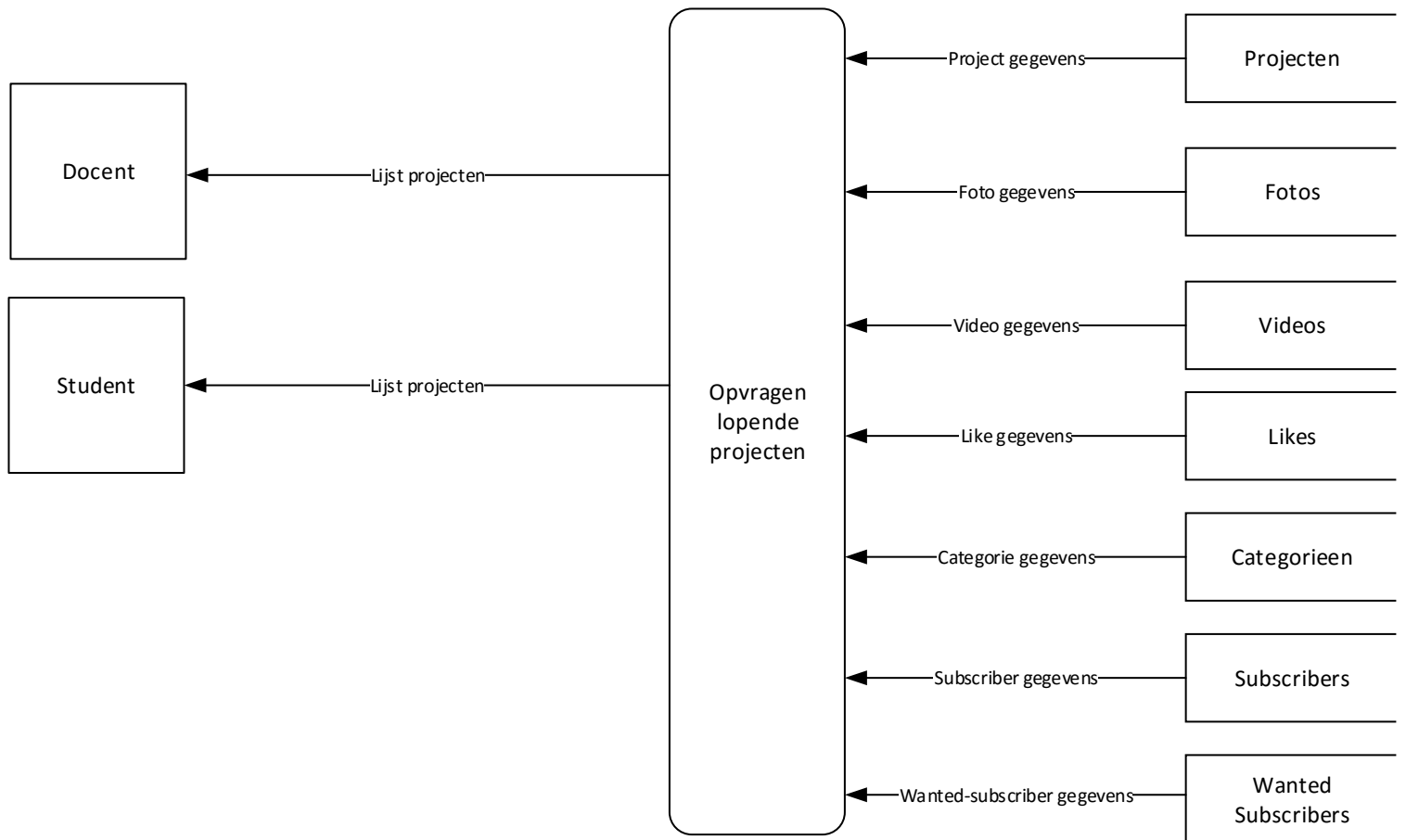
Gasten, docenten en studenten kunnen de detailpagina van een project opvragen door een bepaald project te selecteren uit een lijst.

8. Studenten, docenten en administratoren authenticeren in de applicatie



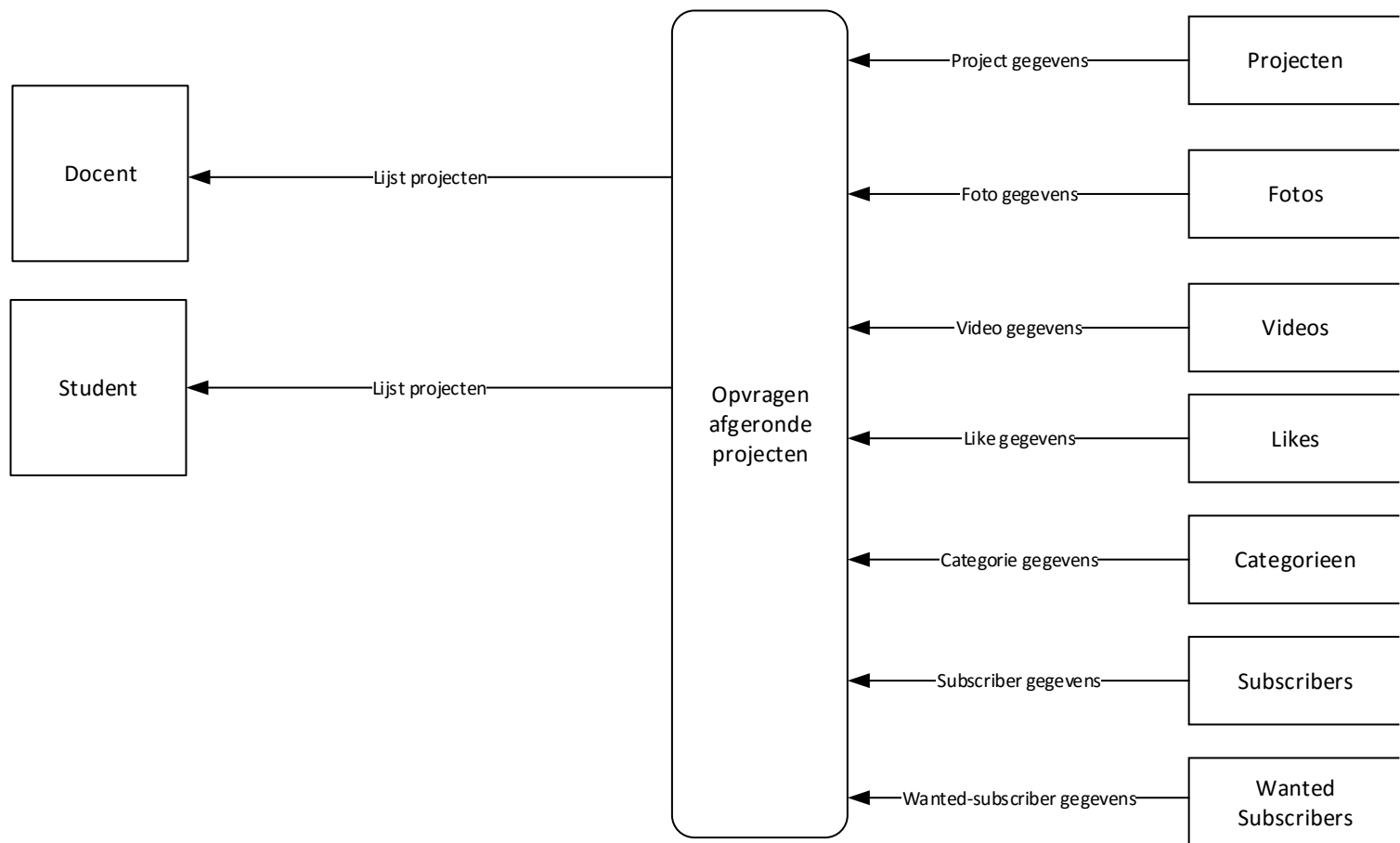
Gebruikers kunnen inloggen om hun rollen te te verifiëren. In een later proces hebben bepaalde gebruikers meer of minder rechten om bepaalde acties uit te voeren en specifieke taken uit te voeren.

9. Docenten, studenten en gasten kunnen lijsten opvragen van lopende projecten



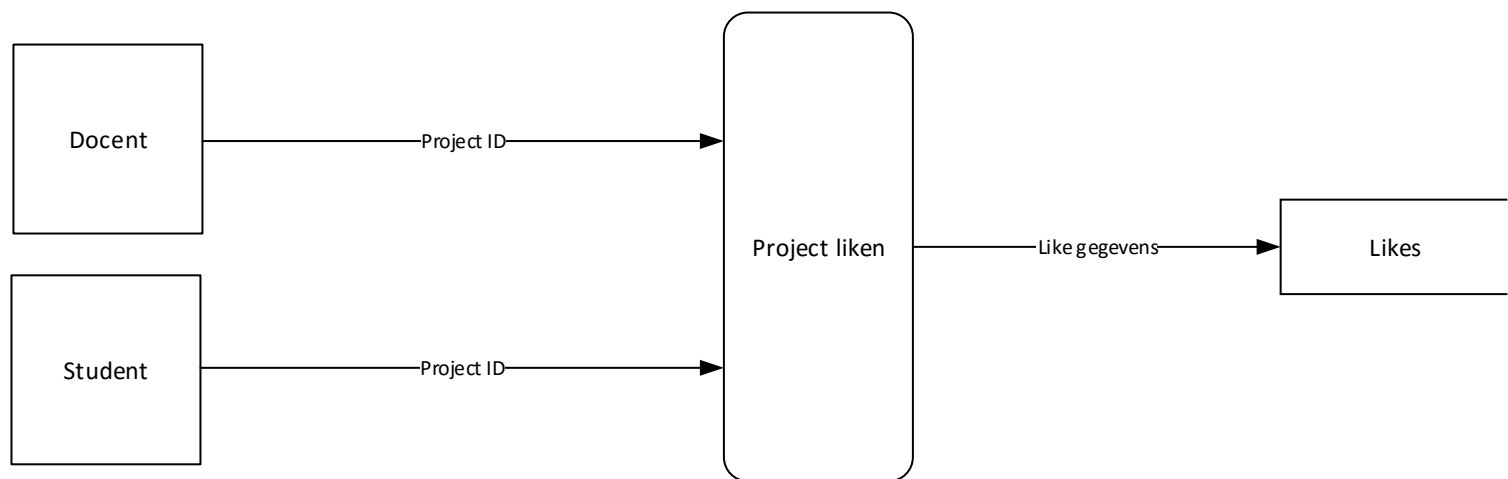
Docenten en studenten kunnen een lijst opvragen van lopende projecten. Dit zijn projecten die geverifieerd zijn en nog niet afgelopen.

10. Docenten, studenten en gasten kunnen lijsten opvragen van afgeronde projecten



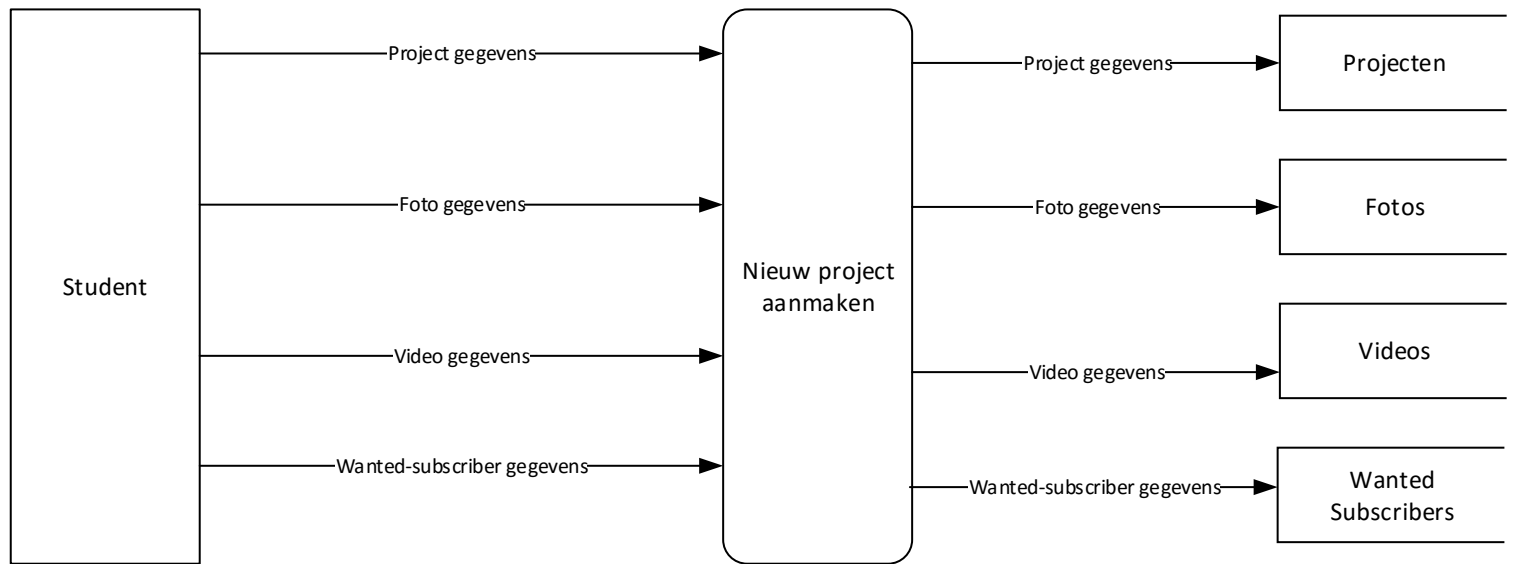
Docenten, studenten en de administator kan de afgeronde projecten bekijken. Niemand kan hierin nog een aanpassing voorzien in vervolgstappen. Het liken van een afgerond project kan wel nog steeds plaatsvinden

11. Docenten en studenten kunnen kunnen projecten liken



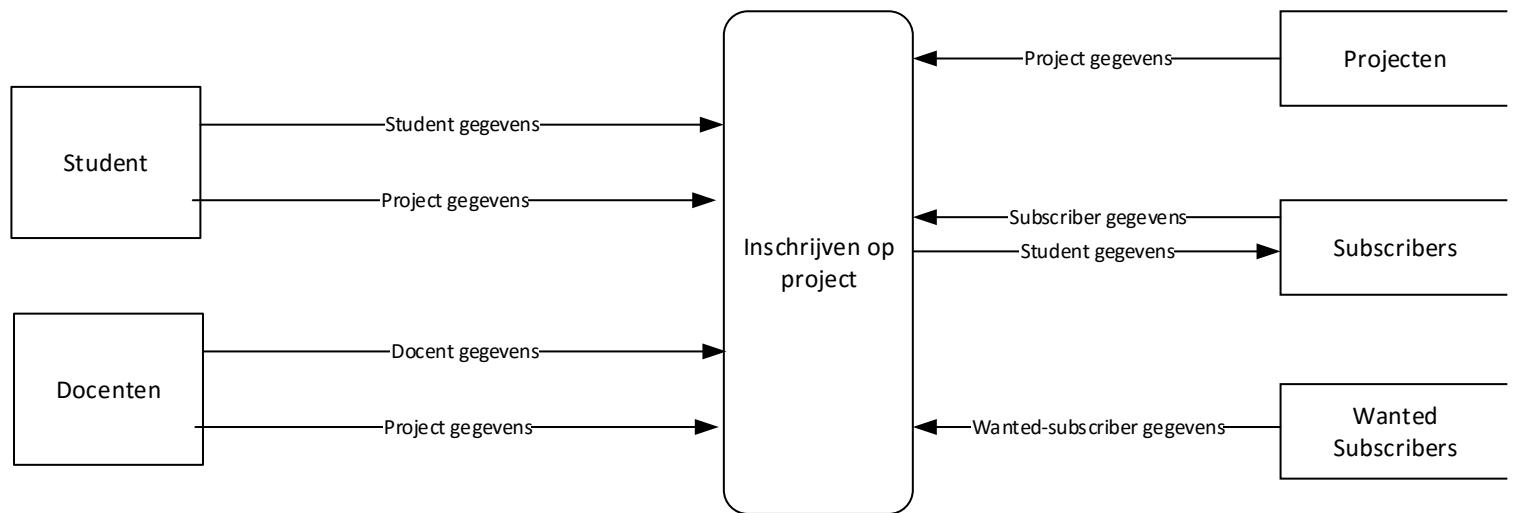
Docenten en studenten kunnen projecten liken indien ze dezelfde actie nog niet hebben voltooid voor hetzelfde project.

12. Studenten kunnen nieuwe projecten aanmaken



Studenten kunnen projecten aanmaken. Tijdens het aanmaken kunnen er gegevens worden ingevoegd zoals foto- en video- locaties. Alsook kan de student subscribers types toevoegen. (bijvoorbeeld: 2 personen van IT, 1 persoon van health)

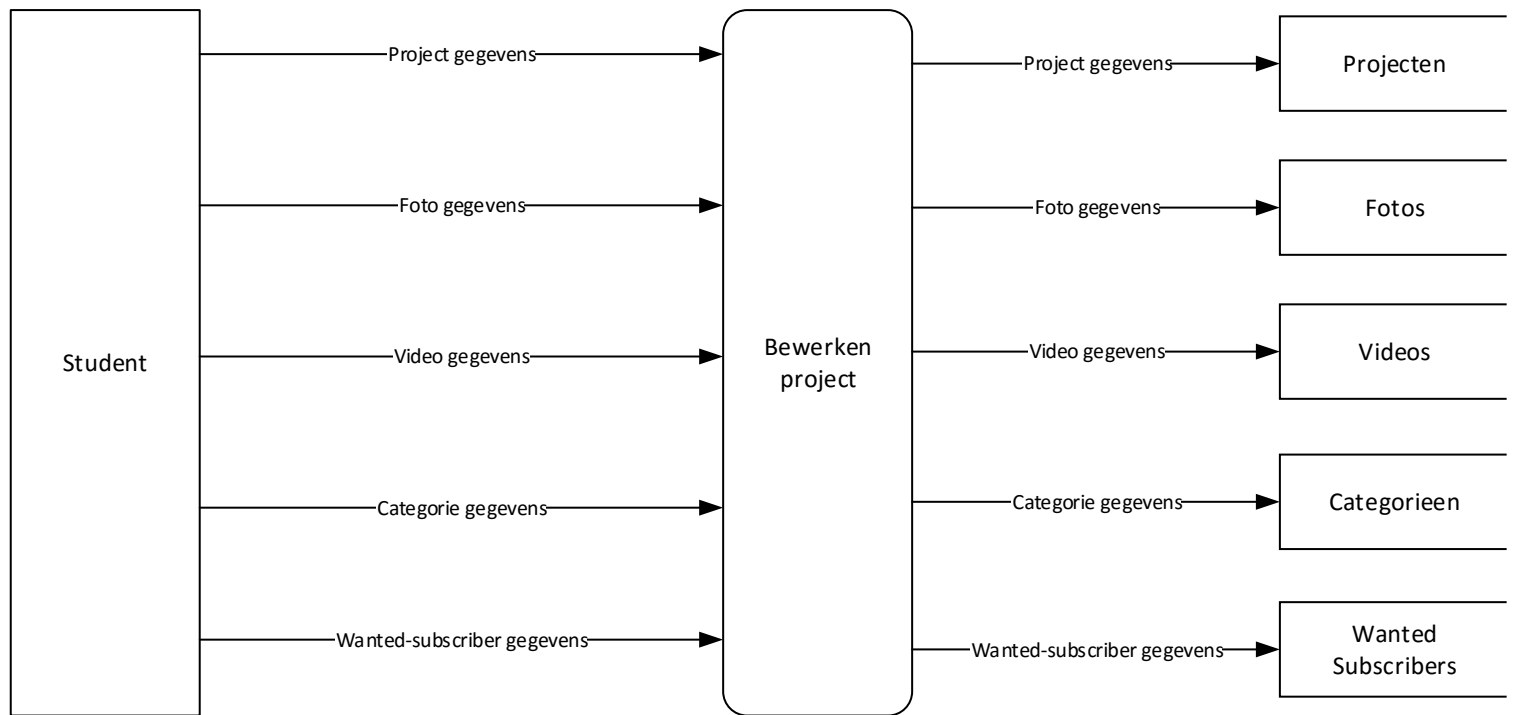
13. Studenten en docenten kunnen inschrijven op projecten



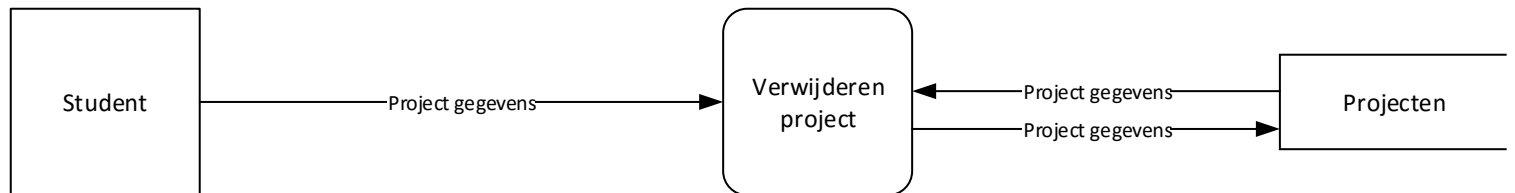
Indien een student zich wenst in te schrijven voor een bepaald project zal het systeem nakijken of de persoon in aanmerking komt (juiste departement, nog niet ingeschreven op een ander project). Wanneer dit het geval is zal het systeem de gebruiker toevoegen als subscriber van het project.

Een docent kan zich ten alle tijde inschrijven op een project. Deze docent telt niet mee tot het aantal gevraagde 'subscribers'. Indien een project door een docent werd gevolgd kan hij op een overzichtspagina de status van verschillende projecten bekijken.

14. Studenten kunnen hun eigen projecten bewerken



15. Studenten kunnen hun eigen projecten verwijderen

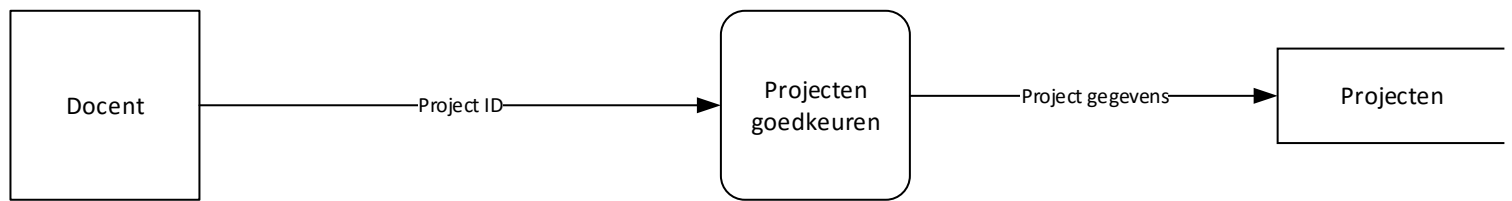


Wijzigingen kunnen enkel doorgevoerd worden indien er nog geen goedkeuring is van een docent.

Studenten kunnen hun eigen projecten enkel verwijderen indien er nog geen goedkeuring is van een docent.

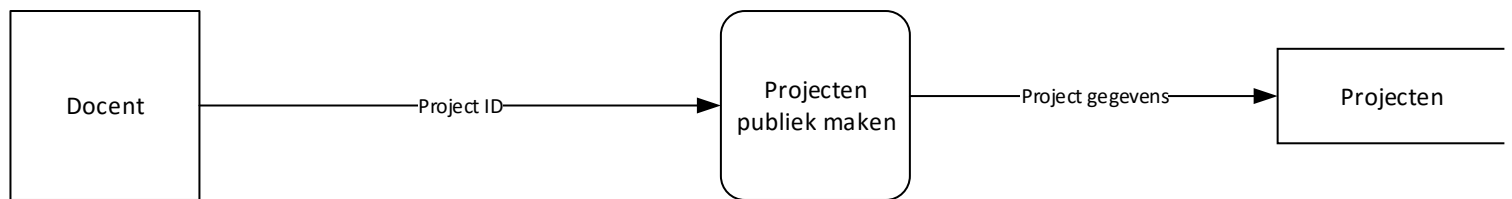
Een project zal nooit verwijderd worden uit de databank, maar zal enkel de status 'verwijderd' krijgen.

16. Docenten kunnen projecten goedkeuren



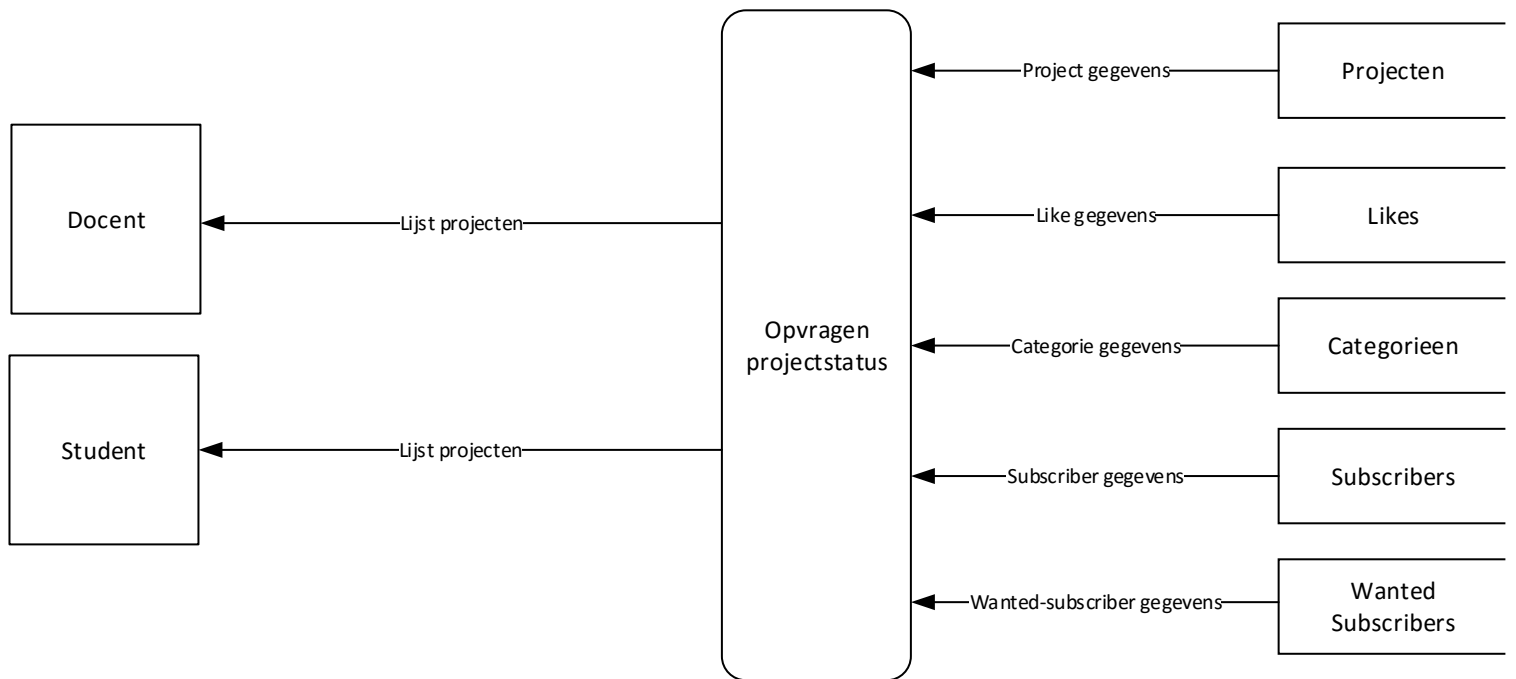
Docenten kunnen projecten goedkeuren. Een goedgekeurd project kan van start gaan op de voorziene datum en kan niet meer gewijzigd / verwijderd worden.

17. Docenten kunnen projecten publiek maken



Docenten kunnen projecten publiek maken.
Een publiek project is ook zichtbaar voor gasten.

18. Studenten en docenten kunnen de status van projecten volgen



Docenten en studenten kunnen de status van hun projecten volgen.

Zowel studenten als docenten kunnen zich inschrijven op meerdere projecten, een overzicht hiervan wordt verkregen via dit proces.

(Bijvoorbeeld: afgelopen dagen zijn er x likes bijgekomen, er zijn zijn x subscribers bijgekomen de laatste week...)

Gegroepeerde gebeurtenissen

A. Beheer iTalent gegevens

1. Administrator definieert categorieën
2. Administrator definieert gebruikers
3. Administrator definieert rollen
4. Administratoren kunnen gebruikers verwijderen

B. Beheer iTalent authenticatie

8. Studenten, docenten en administratoren authenticeren in de applicatie

C. Beheer projectenlijst

5. Docenten, studenten en gasten kunnen een projectenlijst raadplegen
6. Docenten, studenten en gasten kunnen projectenlijst rangschikken
9. Docenten, studenten en gasten kunnen lijsten opvragen van lopende projecten
10. Docenten, studenten en gasten kunnen lijsten opvragen van afgeronde projecten
18. Studenten en docenten kunnen de status van projecten volgen

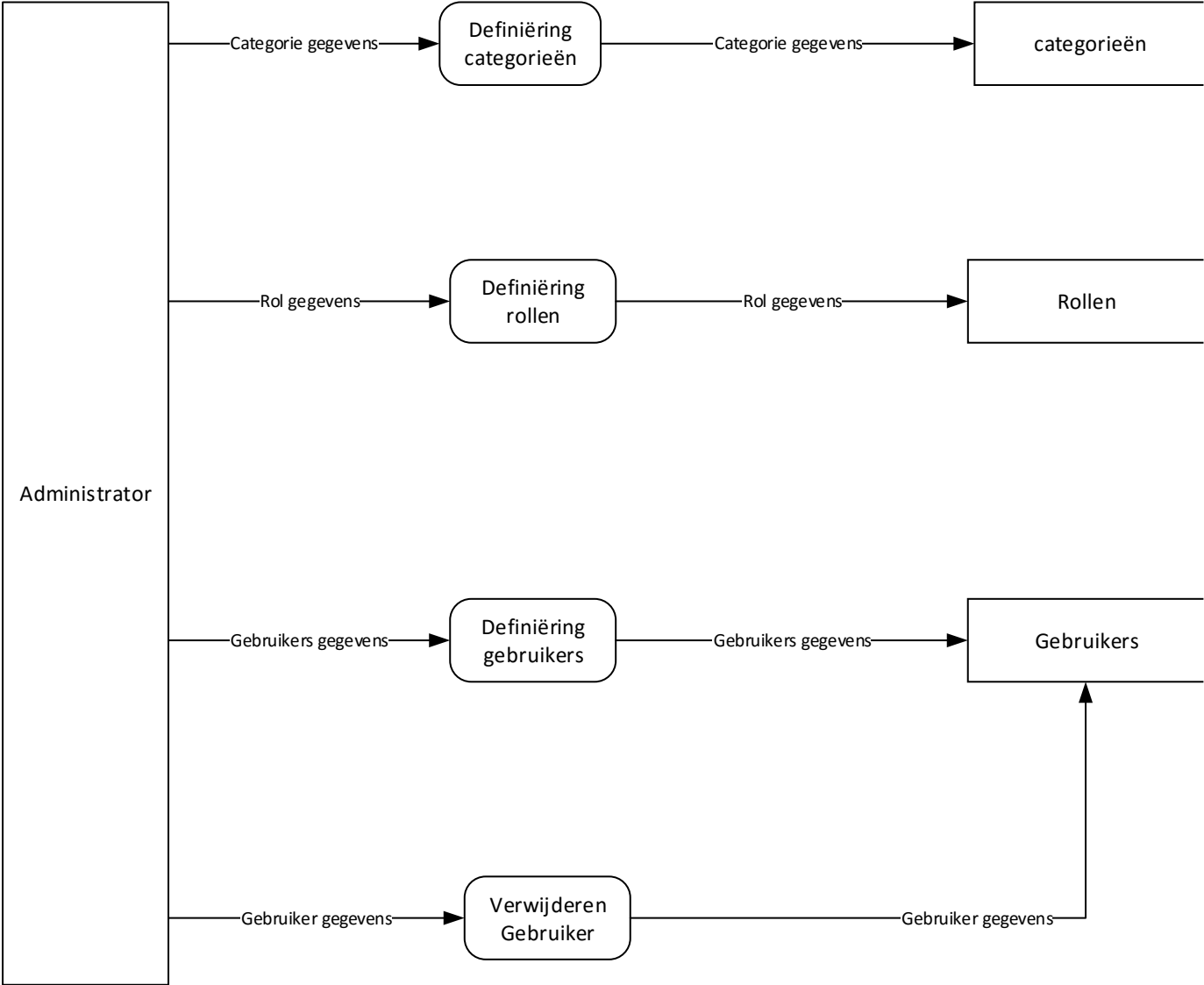
D. Beheer projectdetails

7. Docenten, studenten en gasten kunnen projectdetails opvragen
12. Studenten kunnen nieuwe projecten aanmaken
13. Studenten kunnen inschrijven op projecten
14. Studenten kunnen hun eigen projecten bewerken
15. Studenten kunnen hun eigen projecten verwijderen
11. Docenten en studenten kunnen projecten liken

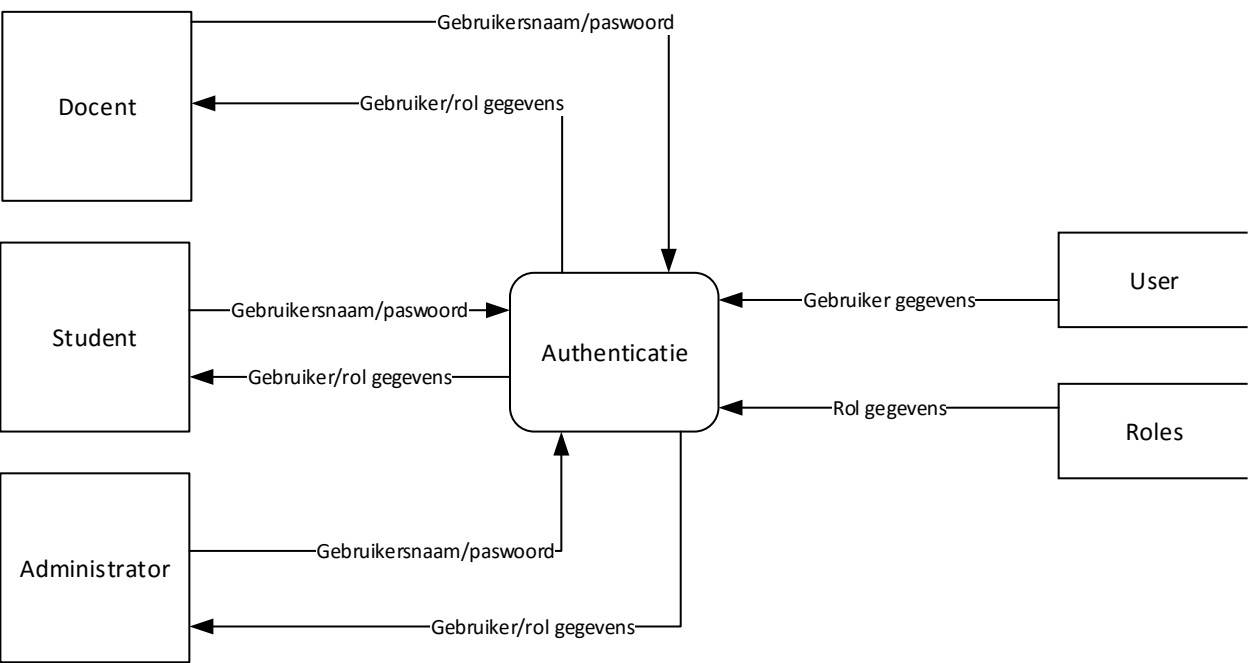
E. Beheer projecttoegang

16. Docenten kunnen projecten goedkeuren
17. Docenten kunnen projecten publiek maken

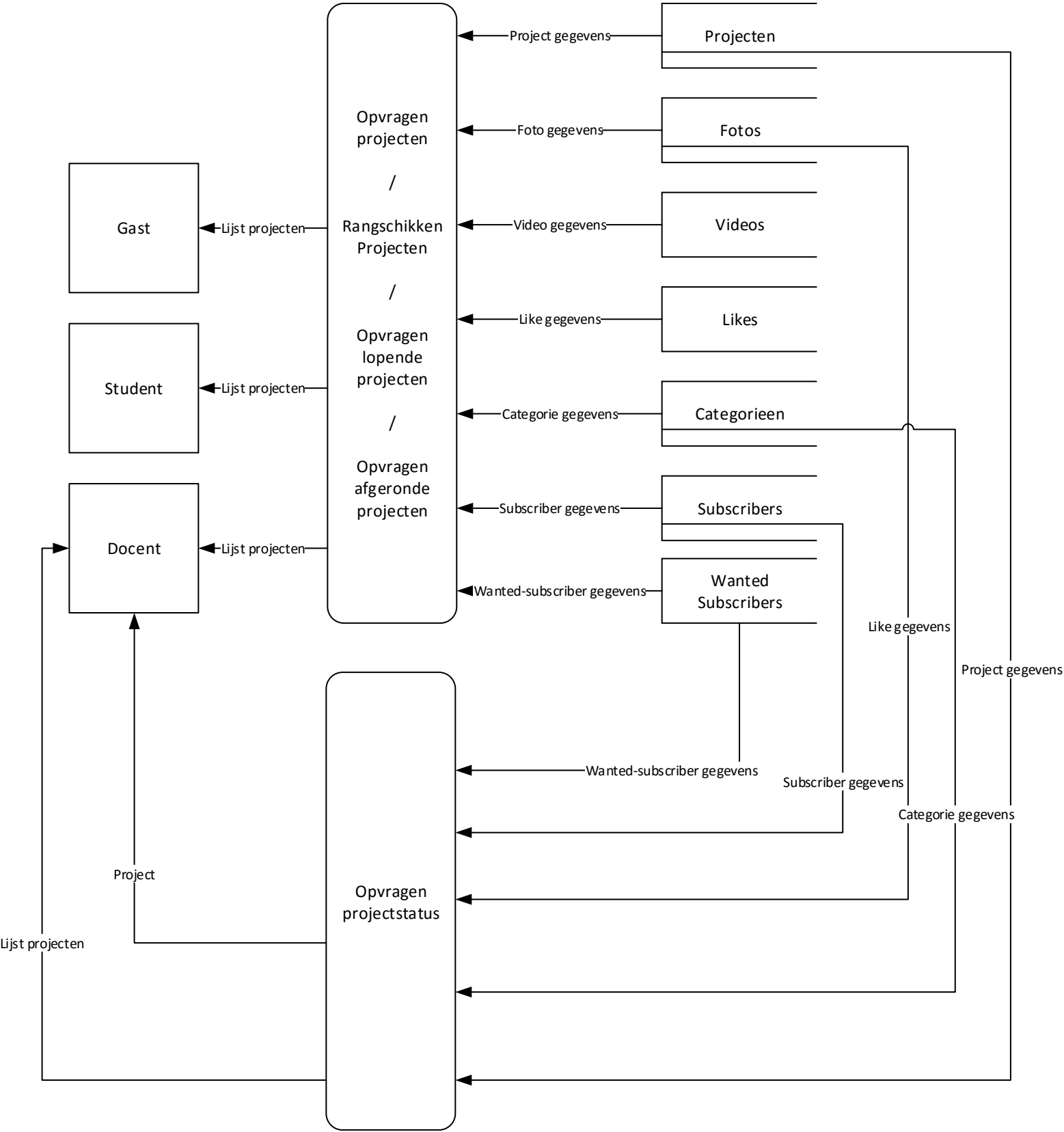
A. Beheer iTalent gegevens



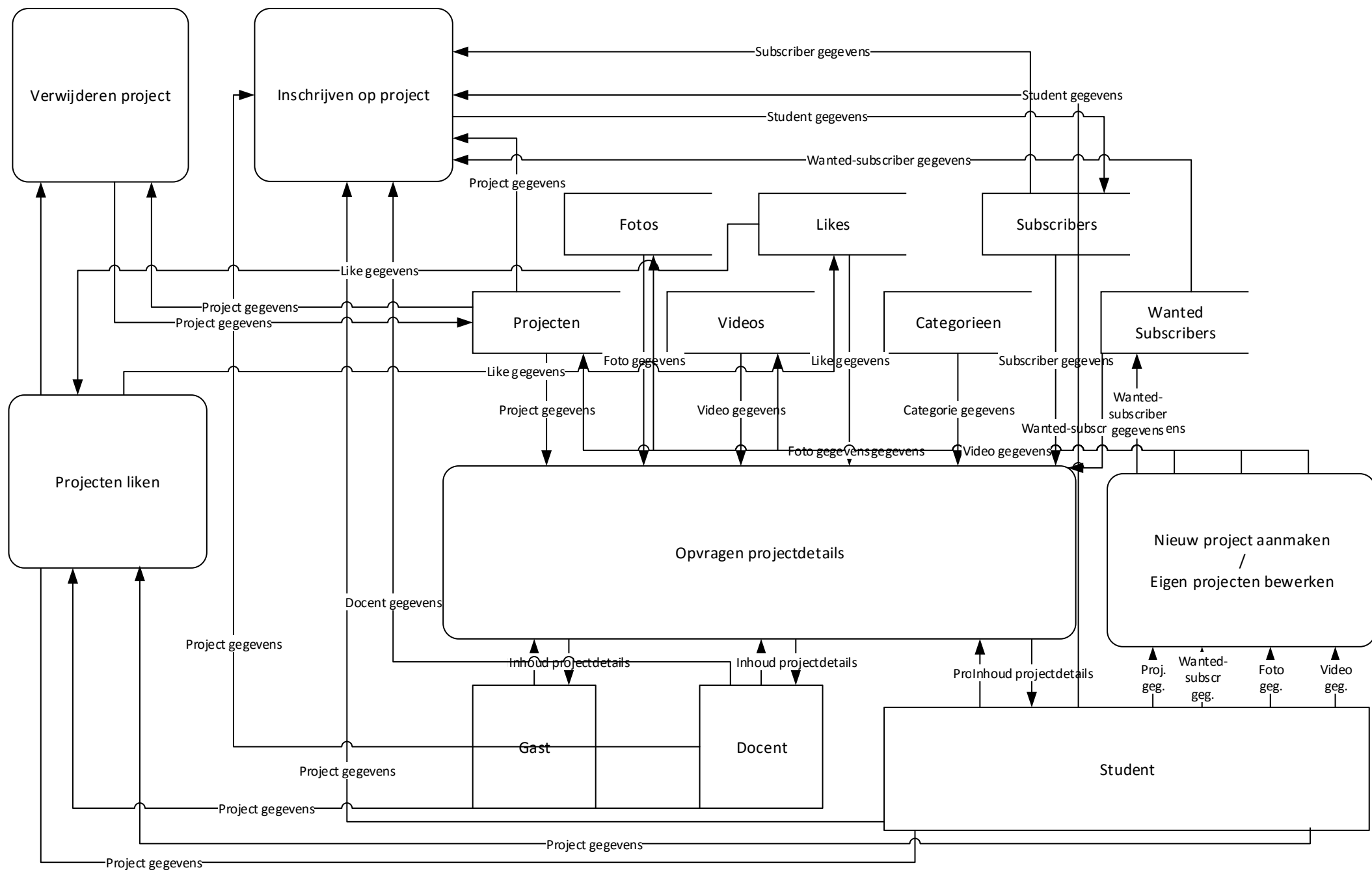
B. Beheer iTalent authenticatie



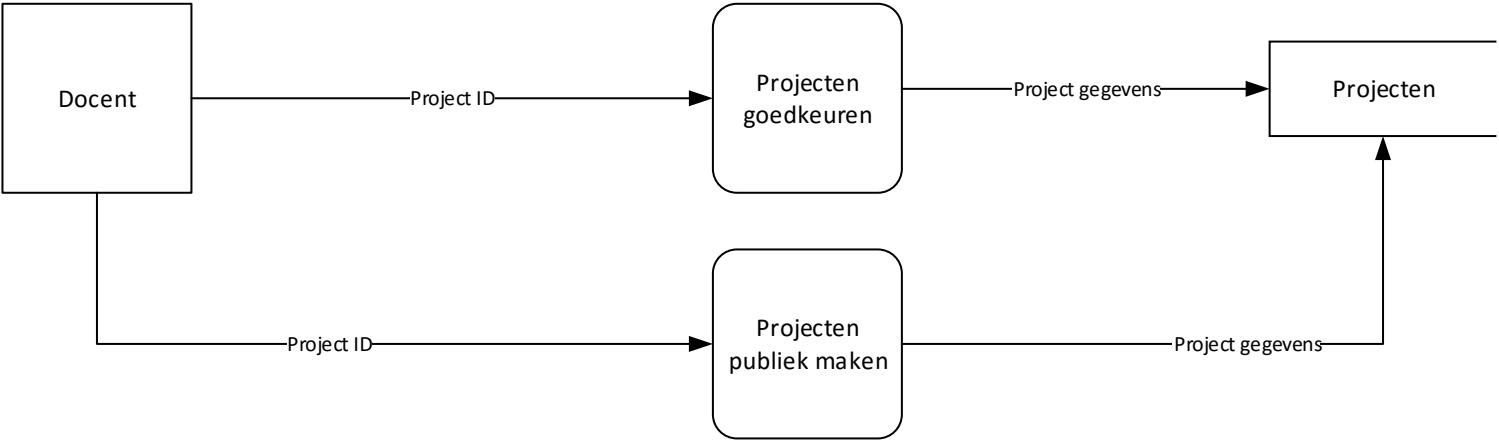
C. Beheer projectenlijst



D. Beheer projectdetails



E. Beheer projecttoegang



Normalisatie

NV0	NV1	NV2	NV3
Gegevens	Klant	Klant	Klant
klantNummer	<u>klantNummer</u>	<u>klantNummer</u>	<u>klantNummer</u>
klantNaam	klantNaam	klantNaam	klantNaam
klantVoornaam	klantVoornaam	klantVoornaam	klantVoornaam
klantStraat	klantStraat	klantStraat	klantStraat
klantStraatNummer	klantStraatNummer	klantStraatNummer	klantStraatNummer
klantPostcode	klantPostcode	klantPostcode	klantEmail
klantLand	klantLand	klantLand	klantTelefoonnummer
klantGemeente	klantGemeente	klantGemeente	postcode_id
klantEmail	klantEmail	klantEmail	
klantTelefoonnummer	klantTelefoonnummer	klantTelefoonnummer	Reservatie
reservatieNummer			<u>reservatieNummer</u>
reservatieBeginDatum	Reservatie	Reservatie	klantNummer
reservatieEindDatum	<u>reservatieNummer</u>	<u>reservatieNummer</u>	plaatsNummer
plaatsNummer	klantNummer	klantNummer	reservatieBeginDatum
plaatsOmschrijving	reservatieBeginDatum	reservatieBeginDatum	reservatieEindDatum
plaatssoortCode	reservatieEindDatum	reservatieEindDatum	
plaatssoortOmschrijving	plaatsNummer	plaatsNummer	Plaats
plaatssoortTarief	plaatsOmschrijving	plaatsOmschrijving	<u>plaatsNummer</u>
kampeerderNummer	plaatssoortCode	plaatssoortCode	plaatsOmschrijving
kampeerderNaam	plaatssoortOmschrijving	plaatssoortOmschrijving	plaatssoortCode
kampeerderVoornaam	plaatssoortTarief	plaatssoortTarief	zoneCode
kampeerdersoort	zoneCode	zoneCode	
kampeerdersoortOmschrijving	zoneOmschrijving	zoneOmschrijving	Plaatssoort
kampeerdersoortTarief			<u>plaatssoortCode</u>
kampeerderDagprijs	Vordering	Vordering	plaatssoortOmschrijving
kampeerderBeginDatum	<u>vorderingNummer</u>	<u>vorderingNummer</u>	plaatssoortTarief
kampeerderEindDatum	klantNummer	klantNummer	
zoneCode	vorderingBedrag	vorderingBedrag	Zone
zoneOmschrijving	vorderingOmschrijving	vorderingOmschrijving	<u>zoneCode</u>
vorderingNummer			zoneOmschrijving
vorderingBedrag	Verblijf	Verblijf	
vorderingOmschrijving	<u>verblijfNummer</u>	<u>verblijfNummer</u>	Vordering
verblijfNummer	klantNummer	klantNummer	<u>vorderingNummer</u>
verblijfBeginDatum	plaatsNummer	plaatsNummer	klantNummer
verblijfEindDatum	verblijfBeginDatum	verblijfBeginDatum	vorderingBedrag
	verblijfEindDatum	verblijfEindDatum	vorderingOmschrijving
	kampeerderNummer	kampeerderNummer	
	kampeerderNaam	kampeerderNaam	Verblijf
	kampeerderVoornaam	kampeerderVoornaam	<u>verblijfNummer</u>
	kampeerderBeginDatum	kampeerderBeginDatum	klantNummer
	kampeerderEindDatum	kampeerderEindDatum	plaatsNummer
	kampeerdersoort	kampeerdersoort	verblijfBeginDatum
	kampeerdersoortOmschrijving	kampeerdersoortOmschrijving	verblijfEindDatum
	kampeerdersoortTarief	kampeerdersoortTarief	
			Kampeerder
			<u>kampeerderNummer</u>
			verblijfNummer
			kampeerderNaam
			kampeerderVoornaam
			kampeerderBeginDatum
			kampeerderEindDatum
			kampeerdersoort
			Kampeerderssoort
			<u>kampeerderssoort</u>
			kampeerdersoortOmschrijving
			KampeerdersoortTarief
			Postcode
			<u>postcode_id</u>
			postcode
			gemeente
			land

Commentaar bij normalisatie

Bij normaalvorm nul (NV0) gaan we kijken welke gegevens we nodig hebben die DFN wil vergaren. Dit gaat van klantgegevens tot reservaties en vorderinggegevens.

Nu we al deze informatie hebben kunnen we deze niet nuttig gebruiken, dus gaan we stappen ondernemen om deze informatie optimaal in tabellen weer te geven. Zoals u al kan zien in normaalvorm 3 (NV3) is dit veel overzichtelijker. Om tot NV3 te gaan moeten we natuurlijk normaalvorm 1 (NV1) en normaalvorm 2 (NV2) eerst bekomen.

Normaalvorm 1:

Om van NV0 naar NV1 te gaan, kijken we waar de herhaalde groepen zich bevinden. We maken aparte groepen hiervan en definiëren daarin een unieke sleutel. Het resultaat is dat we 4 verschillende groepen krijgen nl. Klant, Reservatie, Vordering en Verblijf.

Er zijn meerdere klanten bij DFN. Deze klantgroep moet een unieke waarde hebben om problemen te voorkomen i.v.m. vorderingen, klanten met dezelfde naam, enz. Deze unieke waarde is een klantnummer. De overige klantgegevens worden hier onder weergegeven.

Voor de reservaties maken we best ook een aparte groep aangezien er meerdere reservatie zijn bij DFN, eventueel van dezelfde klant. De unieke sleutel hier is het reservatienummer. Met dit nummer krijgen we een duidelijk overzicht van alle reservaties en het maakt een reservatie uniek. Klantnummer nemen we mee als attribuut om een reservatie te kunnen linken aan één klant. De rest van de attributen die we meenemen in deze groep, zijn de begin- en einddatum van de reservatie, de plaats die gereserveerd wordt en een eventuele zone die de klant reserveert.

De vorderingen moeten we ook apart nemen en uniek identificeren met een vorderingnummer. De reden dat we hiervan een groep maken is omdat er meerdere vorderingen kunnen zijn per klant. Verder worden het klantnummer, het bedrag en de omschrijving van de vordering mee in de groep opgenomen als attributen.

Als laatste groep hebben we de verblijven. Deze groep moeten we creëren, omdat klanten meerdere verblijven kunnen boeken en omdat er zich meerdere kampeerders op één verblijf kunnen bevinden. We lossen dit op met de unieke sleutel verblijfsnummer. Verder nemen we het klantnummer, het nummer van de plaats, de verblijfsdata en de kampeerdergegevens op in deze groep als attributen.

Normaalvorm 2:

Nu we NV1 hebben gaan we kijken wat we verder kunnen doen om naar het einddoel te streven. Om van NV1 naar NV2 te gaan, kijken we of er samengestelde sleutels aanwezig zijn. Dit is niet het geval, dus nemen we het geheel van NV1 over naar NV2.

Normaalkvorm 3:

Onze laatste normaalvorm, NV3, kunnen we enkel verwezenlijken door naar NV2 te kijken en te zien of er groepen zijn met attributen die afhankelijk zijn van velden die niet tot de sleutel behoren. Deze attributen splitsen we af om een nieuwe groep te vormen.

Als we kijken naar de eerste groep, Klant, dan stellen we vast dat we de postcodes apart kunnen weergeven. We maken een groep, Postcode, en voegen een unieke sleutel toe. Deze sleutel vermelden we in de groep waar het oorspronkelijk weergegeven stond. We plaatsen hier dus postcode_Id als attribuut bij de groep Klant.

Als we kijken naar de groep reservatie, stellen we vast dat we plaatsen ook apart kunnen nemen aangezien een plaats niet altijd vasthangt aan een reservatie. Hier doen we hetzelfde als bij postcode en we maken een nieuwe groep aan. Deze noemen we Plaats en geven deze de unieke sleutel, plaatsnummer. Opgelet, want we nemen hier eigenlijk 4 attributen mee, namelijk: plaatsomschrijving, plaatssoortcode, zonecode en zoneomschrijving. Maar omdat we zonecode en zoneomschrijving verder kunnen afzonderen, splitsen we deze onmiddellijk op. Hier maken we een groep Zone van, met een unieke sleutel zonecode. We nemen zoneomschrijving mee als attribuut.

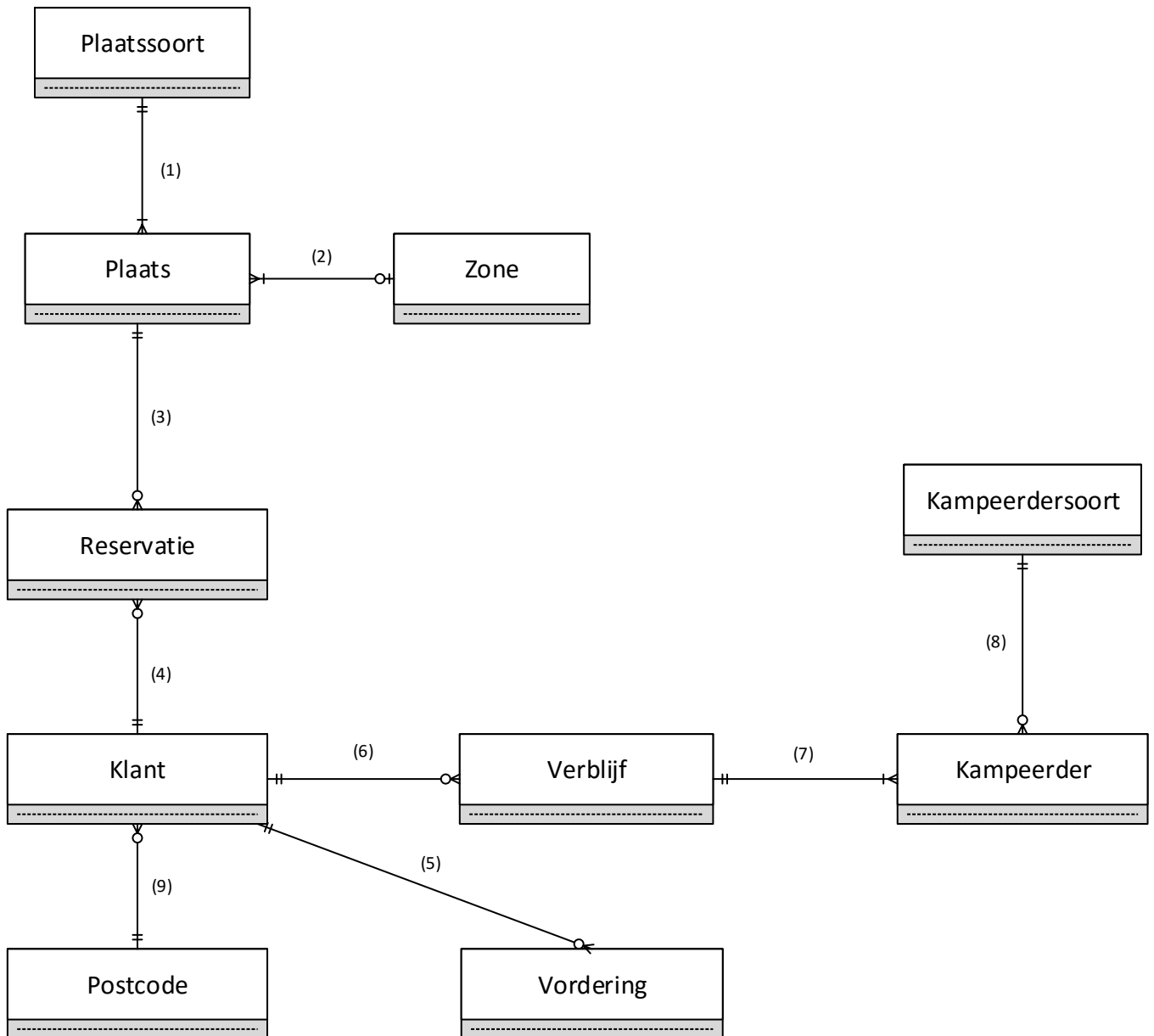
De groep vordering, blijft ongewijzigd omdat alle attributen afhankelijk zijn van de sleutel.

Als we naar de laatste ongewijzigde groep Verblijf kijken, stellen we vast dat we deze kunnen splitsen. De kampeerdergegevens staan op zich en hier maken we een aparte groep van. In de groep Verblijf staan nu nog enkel attributen die te maken hebben met het verblijf. Deze zijn: de klant, de plaats en van wanneer tot wanneer deze geboekt is.

Voor de kampeerdergegevens maken we de groep Kampeerder aan en maken een unieke sleutel aan met het nummer van de kampeerder. Verder nemen we de gegevens van de kampeerdereen, samen met het verblijfsnummer en de sleutel van de groep Kampeerdereen. Ook hier doen we hetzelfde als bij plaats en plaatssoort. We kunnen de Kampeerder verder splitsen, namelijk de attributen kampeerdereen, de omschrijving en het tarief steken we in een aparte groep, Kampeerdereen.

Door deze stappen toe te passen, komen we bij de derde normaalvorm en kunnen we er van uit gaan dat met deze groepen, de gegevens veilig zijn voor veranderingen in het systeem van DFN.

ERD



Commentaar ERD

1) Plaatssoort – Plaats

Plaatssoorten zijn specifieke plaatsen op de camping. Een klant kan bv. kiezen of hij aan het water wil reserveren. Plaatsen omvat het algemene aspect. Het is dus voor deze associatie duidelijk dat er zich meerdere plaatsen in één plaatssoort kunnen bevinden. Anderzijds kan één plaats zich maar in één plaatssoort bevinden.

2) Plaats - Zone

Als een klant meerdere plaatsen reserveert, dit gebeurt meestal voor grote groepen, zal DFN een grote zone ter beschikking stellen voor deze groep van klanten. Een zone is dus minstens één plaats, maar in de meeste gevallen meerdere plaatsen, daarom deze associatie. Een plaats daarentegen kan zich oftewel in een zone bevinden of niet. Het is niet mogelijk dat meerdere zones dezelfde plaats bevatten. Dit zou voor conflicten zorgen.

3) Plaats - Reservatie

Een plaats dient om ingenomen te worden door een klant. Dit kan door een reservatie. Als deze plaats gereserveerd is, kan niemand anders deze plaats innemen voor de gereserveerde datum. Daarom kan er ook maar één plaats zijn voor één reservatie. Zoals eerder vermeldt, kan een plaats al dan niet gereserveerd zijn. Daarom is de associatie van plaats naar reservatie optioneel en/of meerdere reservaties. Een plaats kan niet verhuurd worden, maar kan langs de andere kant ook 5 maal geboekt zijn. Verspreidt over een heel jaar is dit mogelijk.

4) Reservatie - Klant

Bekeken vanuit het standpunt van de klant, kan deze niet reserveren, één keer reserveren of meerdere reserveringen doen bij DFN. Daarom de optionele/meerdere associatie vanuit de klant richting reserveringen. Een reservatie daarentegen kan maar horen tot één klant. Het is ook één en enkel één klant per reservatie, want er bestaat geen reservatie zonder een klant, en als je meerdere klanten op één reservering gaat zetten zit je met een groot conflict. (Een klant kan bestaan uit meerdere mensen van hetzelfde gezin/groep).

5) Klant - Vorderingen

Vorderingen zijn betalingen vanaf de reservatie tot na het verblijven zelf (achterstallige rekening). Een vordering kan maar tot één klant behoren. Het is niet de bedoeling dat een vordering van één klant verdeeld wordt over andere klanten die niets met die vordering te maken hebben. Een vordering is langs de andere kant ook optioneel. Niet iedereen reserveert op voorhand of niet iedereen betaalt na het vertrek. Maar een klant kan wel meerdere vorderingen hebben. Bv.: een klant reserveert, en kan na vertrek niet direct betalen. DFN zal deze vorderingen controleren.

6) Klant - Verblijf

Een verblijf kan maar aan één klant gelinkt worden. Deze omvat alle gegevens van het verblijf op één moment voor die ene klant. Een klant heeft natuurlijk de optie om te verblijven bij DFN, en heeft dus ook de optionele associatie en de keuze om meerdere verblijven te boeken. De Klant heeft ook de keuze om de samenstelling van de groep en zijn locatie aan te passen, dus heeft de klant meerdere verblijven op zijn naam staan en komt dit ook zo op de factuur te staan.

7) Verblijf - Kampeerder

De data store Kampeerder omvat alle gegevens van de kampeerders ingeschreven onder één klant. Klanten kunnen meerdere verblijven op hun naam hebben staan (wijziging locatie/groep), maar bevindt zich steeds maar op één plaats op één bepaald tijdstip, daarom zijn de kampeerdergegevens op één tijdstip maar gelinkt aan één verblijf. Indien er een verblijf is geboekt, zijn er dus ook klanten (kampeerders) die zich op een gegeven moment daar zullen bevinden. Aan deze klanten hangen de kampeerdergegevens, die dus bij minstens één verblijf toebehoren.

8) Kampeerder - Kampeerdersoort

De kampeerdersoort omvat over welke kampeerdere we het hebben. Hebben we het over dieren, kinderen (tienaers) of volwassenen. Dit heeft te maken met hoe DFN de prijsberekening maakt.

Daarom kan een kampeerder dus maar één soort kampeerder zijn. Een kind valt onder de categorie kinderen, niet onder kinderen én volwassenen. Een kampeerdersoort heeft niet altijd zijn mensen aanwezig. Bv.: Een klant (gezin) gaat op vakantie bij DFN en heeft 2 volwassenen en 1 kind. De klant heeft geen dier meegenomen naar de camping en er zal dan ook geen soort, dieren, aangemaakt worden voor deze klant. Zoals aangehaald met het voorbeeld is het dus duidelijk dat er meerdere kampeerdere zijn per soort(2 volwassenen).

9) Klant - Postcode

Om snel en correct gegevens van de klant te kunnen achterhalen, maken we een data store Postcode aan. Vanuit het standpunt van de klant kunnen we vaststellen dat elke klant één postcode heeft. Een klant heeft maar één officieel verblijfsadres en kan dus maar één postcode meegeven. Een postcode kan andersom wel meerdere klanten omvatten. Mijn buur en ik wonen in dezelfde gemeente (dus dezelfde postcode) en wij hebben de optie om bij DFN te boeken, wij zijn dus twee verschillende klanten, maar zijn niet verplicht te verblijven bij DFN. Daarom dus de optionele en meerdere associatie tussen postcode en klant.

Logisch Gegevensmodel

Klant	
<u>klantNummer</u> : Integer(7)	
klantNaam: String (2..30)	
klantVoornaam: String (2..30)	
klantStraat: String (2..50)	
klantStraatNummer: String(15)	
klantEmail: Email	
klantTelefoonnummer: String (8..15)	
postcodeId: Integer(5)	=> Postcode(postcodeId)
Reservatie	
<u>reservatieNummer</u> : Integer(7)	
klantNummer: Integer(7)	=> Klant(klantNummer)
plaatsNummer: Integer(5)	=> Plaats(plaatsNummer)
reservatieBeginDatum: Datum	
reservatieEindDatum: Datum	
Plaats	
<u>plaatsNummer</u> : Integer (5)	
plaatsOmschrijving: String(250)	
plaatssoortCode: String (7)	=>Plaatssoort(plaatssoortCode)
zoneCode: String(10)	=>Zone(zoneCode)
Plaatssoort	
<u>plaatssoortCode</u> : String (7)	
plaatssoortOmschrijving: String(250)	
plaatssoortTarief: Euro	
Zone	
<u>zoneCode</u> : String(10)	
zoneOmschrijving: String(250)	
Vordering	
<u>vorderingNummer</u> : Integer(7)	
klantNummer: Integer(7)	=> Klant(klantNummer)
vorderingBedrag: Euro	
vorderingOmschrijving: String(250)	
Verblijf	
<u>verblijfNummer</u> : Integer(5)	
klantNummer: Integer(7)	=> Klant(klantNummer)
plaatsNummer: Integer (5)	=> Plaats(plaatsNummer)
verblijfBeginDatum: Datum	
verblijfEindDatum: Datum	
Kampeerder	
<u>kampeerderNummer</u> : Integer(7)	
verblijfNummer: Integer(5)	=> Verblijf(verblijfNummer)
kampeerderNaam: String (2..30)	
kampeerderVoornaam: String (2..30)	
kampeerderBeginDatum: Datum	
kampeerderEindDatum: Datum	
kampeedersoort: String (3)	=> Kampeedersoort(kampeedersoort)
Kampeedersoort	
<u>kampeedersoort</u> : String (3)	
kampeedersoortOmschrijving: String(250)	
kampeedersoortTarief: Euro	
Postcode	
<u>postcodeId</u> : Integer(5)	
postcode: String(8)	
gemeente: String(58)	
land: String(36)	

Commentaar logisch gegevensmodel

Om het logisch gegevensmodel even kort toe te lichten bekijken we entiteit per entiteit.

We zijn dit logisch gegevensmodel bekomen door de derde normaalvorm (zie pagina 24) verder te verwerken met datatypes.

Als we de entiteit **Klant** nader bekijken, zien we enkele integers, strings en een email datatype(s).

Wij hebben ervoor gekozen om een string toe te passen op klantStraatNummer zodat we ook het busnummer met de tekst 'bus' kunnen toevoegen. We hebben voor de entiteit Klant gekozen om met een abstract data type te werken: Email. Dit datatype wordt toegepast om een emailadres in op te slaan. Het emailadres moet een '@'-teken en een punt bevatten. De foreign key die we hier gebruiken, postcodeId, verwijst naar de primary key postcodeId, van de entiteit Postcode.

Bij de entiteit **Reservatie** hebben we Datum gebruikt als datatype om de reservatiedata in op te slaan.

Het data type Datum omvat de datum zonder tijdstip. Ook hier hebben we foreign keys toegepast om een link te leggen met 2 andere entiteiten: Klant en Plaats. In dit geval linken we uit Reservatie: klantNummer, met klantNummer uit de entiteit klant en plaatsNummer uit de entiteit Plaats.

In de entiteit **Plaats** gebruiken primaire (niet zelf-gedefinieerde) datatypes. Bij plaatsOmschrijving hebben we voor 250 characters gekozen zodat hier een ruime beschrijving toegevoegd kan worden. Ook hier zijn er foreign keys: plaatssoortCode verwijst naar de primary key van Plaatssoort, en zoneCode naar die van Zone. Zo is de entiteit Plaats in verband gebracht met deze 2 entiteiten.

Plaatssoort heeft een abstract datatype: Euro. Euro definiëren we als decimale waarde, met een maximum van 7 cijfers voor en 2 cijfers na de komma.

Over de entiteit **Zone** valt niets nieuws te vertellen. De primary key heeft als type string(10).

De 10 characters komen natuurlijk overeen met de primary key van de entiteit Plaats.

In de entiteit **Vordering** gebruiken we een foreign key klantNummer, die zo het verband maakt met de entiteit Klant. Ook gebruiken we een van de abstracte datatypes: Euro (zie Plaatssoort).

In de entiteit **Verblijf** hebben we bewust gekozen om het verblijfsNummer als integer(5) te definiëren. Niet meer dan 5, omdat we nooit meer verblijven kunnen hebben op een moment dan er plaatsen ter beschikking zijn. Ook hier hebben we 2 foreign keys: klantNummer verwijst naar de entiteit Klant, plaatsNummer naar de entiteit Plaats.

De entiteit **Kampeerder** verbindt zich met de entiteiten Kampeederssoort en Verblijf via 2 foreign keys. Het veld Kampeederssoort verbindt Kampeerder met de entiteit Kampeederssoort en het veld verblijfNummer met de entiteit Verblijf.

De entiteit **Kampeederssoort** gebruikt het abstracte data type Euro (zie Plaatssoort). De omschrijving kan maximaal 250 tekens bevatten.

Als laatste entiteit bespreken we **Postcode**. De datatypes voor de attributen hebben we precies kunnen specificeren. Een Belgische postcode heeft 4 cijfers, een Nederlandse heeft 4 cijfers en 2 letters, andere landen in Europa gaan nog verder. Dit lossen we op met een string van maximaal 8 tekens. Bij gemeente tellen we een string van maximaal 58 tekens, we hebben hiervoor gekozen omdat de plaatsnaam met het meest aantal tekens, 58 tekens bevat. Voor land gebruiken we hetzelfde principe: het land met de langste naam bevat 36 tekens.