



Individuele Opdracht ASP.NET

JEROEN ROOVERS

Jeroen Roovers | S23 | 8 januari 2016

Inhoudsopgave

Versiebeheer	3
Opdrachtoomschrijving	4
Planning	5
User Requirement Specifications	6
Inleiding	6
Functionele eisen (MoSCoW)	6
Use Case Diagram	7
Use Cases	8
Acceptatie Test Plan	11
Inleiding	11
Acceptatiecriteria	11
Testmatrix	11
Acceptatie Test plan	12
Conclusie	13
Ontwerp	14
Inleiding	14
Entiteit-Relatie Diagram	15
ERD toelichting	Fout! Bladwijzer niet gedefinieerd.
Database Ontwerp	16
DBO toelichting	Fout! Bladwijzer niet gedefinieerd.
Domeinmodel	17
Domeinmodel Toelichting	17
Klassendiagram	18
Klassendiagram Toelichting	19
Infrastructuur	20

Versiebeheer

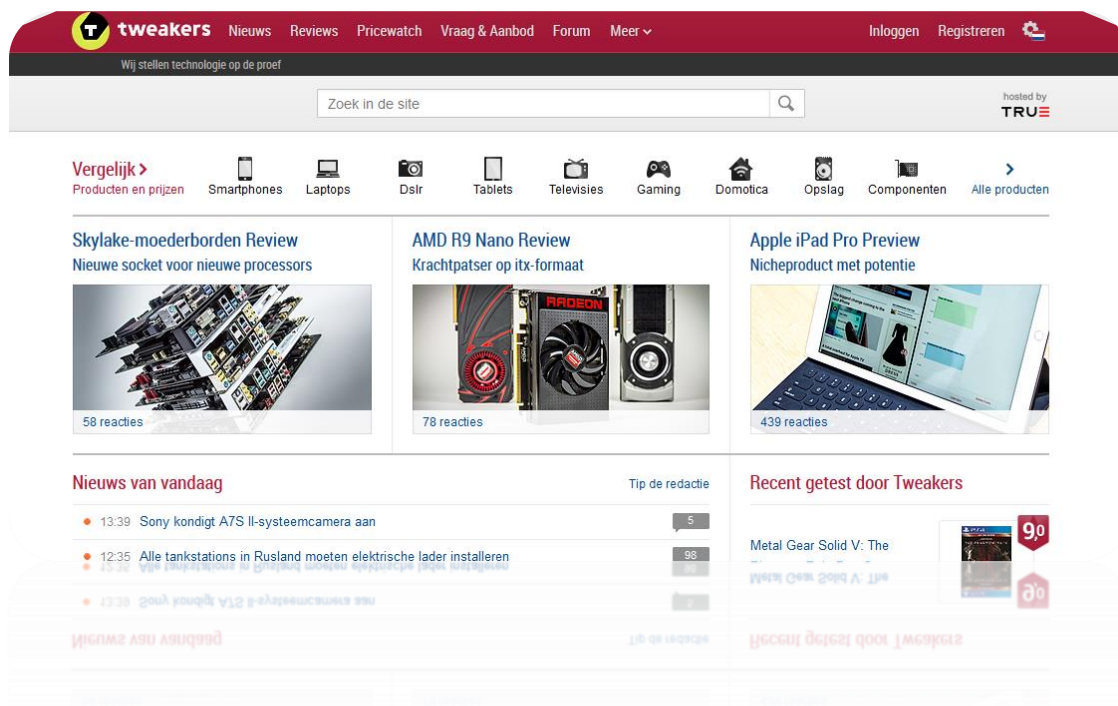
Hier wordt de versie van het document bijgehouden. In de tabel staan van links naar rechts de versie, de datum waarop de aanpassingen gedaan zijn, en een omschrijving van de aanpassingen.

Versie	Datum	Omschrijving
1.0	25-11-2015	Eerste versie: opdrachtomschrijving
1.1	3-12-2015	Analyse Toegevoegd
1.2	4-12-2015	Ontwerp Toegevoegd
1.3	9-12-2015	Domeinmodel Toegevoegd
1.4	6-1-2015	MoSCoW Toegevoegd
1.5	7-1-2015	Inleidingen en Acceptatiecriteria en Infrastructuur Toegevoegd
1.6	8-1-2015	Herschreven naar formaat zoals verwacht bij beoordeling (URS, ATP, Ontwerp). Klassendiagram ingevoegd vanuit applicatie.
2.0	8-1-2015	Testplan uitgevoerd

Opdrachtomschrijving

Voor deze opdracht ga ik een applicatie maken die gebaseerd is op de website van Tweakers.net (website: www.tweakers.net). Tweakers is een nieuwssite die zich voornamelijk richt op technologie. Naast de standaard nieuwsberichten, publiceert Tweakers regelmatig uitgebreide artikelen waarin ze dieper op een bepaald onderwerp ingaan, of een product reviewen.

Gebruikers kunnen op de site inloggen om reacties te plaatsen onder nieuwsberichten en reviews. Ook heeft de site een prijsvergelijker waarin je producten, per categorie, kunt filteren en met elkaar vergelijken. Ten slotte is er nog een forum aanwezig op de website, waar de ingelogde gebruikers topics kunnen aanmaken en wederom kunnen reageren.



De database gaat gebruikt worden om alle producten, prijzen, winkels, artikelen (inclusief reviews), gebruikersreviews, reacties en gebruikersaccounts bij te houden.

Klassen die ik minimaal nodig ga hebben om deze applicatie op te bouwen zijn als volgt:

- Database-klasse
- Winkel
- Prijs
- Product
- Review
 - Reviewpagina
- Nieuws
- Reactie
- Gebruikersreview
- Useraccount

Functionaliteit die ik ga implementeren zijn o.a.

- Inloggen en registreren voor gebruikers
- Het lezen en zoeken van nieuwsartikelen
- Het lezen en zoeken van reviews
- Het zoeken van producten
- Details opvragen van een product inclusief prijzen en bijbehorende winkel
- Het plaatsen van productreviews door gebruikers
- Het plaatsen van reacties door gebruikers

Functionaliteit die wel op de site aanwezig is en die ik **niet** ga implementeren:

- Het forum
- 2^ehands marktplaats
- Nesting van reacties (reacties op een reactie op een reactie)
- Vacature
- Video
- Benchmarks

Planning

Op basis van de geplande functies heb ik de volgende planning opgesteld:

Week	Activiteiten	Uren	Klaar?
11	Omschrijving	2	✓
12	Analyse en Ontwerp	8	✓
13	Implementatie hoofdpagina	16	✓
14	Implementatie login & bekijken van nieuws en reviews	16	✓
15	Implementatie Prijsvergelijker	8	✓
16	Implementatie Productreviews	8	✓
17	Afronding	16	~

User Requirement Specifications

Inleiding

In de URS stel ik vast welke zaken aan bod komen bij het uitvoeren van de opdracht. Ik heb hiervoor de gestelde eisen in de omschrijving omgezet basale functies met hoge prioriteit. Overige zaken die logischerwijs gewenst zijn, maar eigenlijk voortbouwen op de functie met hoge prioriteit, krijgen een lagere prioriteit. Vervolgens behandel ik de interactie tussen de gebruikers en de functies van het systeem. Deze zijn weergegeven als een overzicht in de vorm van een diagram alsook uitgewerkt per functie.

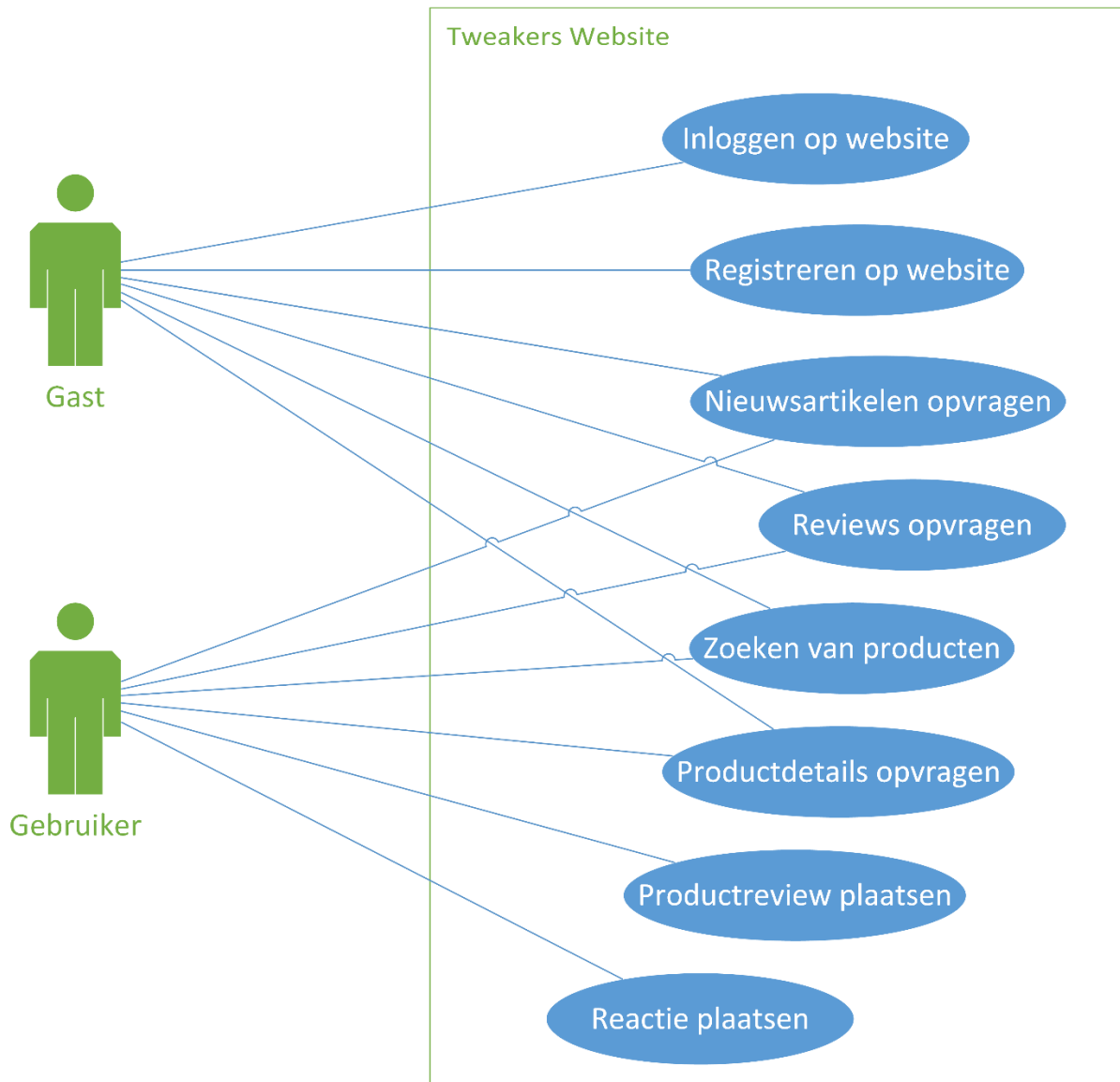
Functionele eisen (MoSCoW)

Ik heb de volgende eisen opgesteld op basis van de opdrachtomschrijving. Alles requirements met prioriteit M worden bovendien uitgewerkt in het ATP en de user requirements specificaties.

Term	Functie	Prioriteit
Inloggen	De bezoeker kan inloggen op de website	<i>Must</i>
Registreren	De bezoeker kan registreren op de website	<i>Must</i>
Nieuws Bekijken	De bezoeker kan nieuws bekijken	<i>Must</i>
Review Bekijken	De bezoeken kan reviews bekijken	<i>Must</i>
Product Zoeken	De bezoeker kan producten zoeken op naam	<i>Must</i>
Productdetails Opvragen	De bezoeken kan productdetails opvragen van een product	<i>Must</i>
Review Plaatsen	Een gebruik kan gebruikersreviews plaatsen bij een product	<i>Must</i>
Reactie Plaatsen	Een gebruiker kan reacties plaatsen bij een nieuwsartikel, review of gebruikersreview	<i>Must</i>
Nieuws Zoeken	Bezoeker kan ik een zoekbalk op naam naar nieuws zoeken	<i>Should</i>
Reviews zoeken	Bezoeker kan ik een zoekbalk op naam naar reviews zoeken	<i>Should</i>
Forum	Gebruikers kunnen het forum bekijken en topics plaatsen	<i>Won't</i>
Vraag & Aanbod	Gebruikers kunnen zoeken op advertenties van 2 ^e hands artikelen, artikelen te koop zetten, en reacties plaatsen onder advertenties	<i>Won't</i>
Reactie Nesting	Gebruikers kunnen reageren op reacties	<i>Won't</i>
Vacatures	Bezoekers kunnen vacatures bekijken	<i>Won't</i>
Videomateriaal	Nieuws en reviews bevatten videomateriaal	<i>Won't</i>
Benchmarks	Bezoekers kunnen benchmarks bekijken	<i>Won't</i>

Use Case Diagram

Hier leg ik uit hoe gebruikers van het systeem interactie aan kunnen gaan. Omdat het een soort van nieuws-site betreft, kunnen zowel niet-registreerde gasten en gebruikers een groot deel van de website bezoeken. Wanneer je niet registreert bent, kun je bovendien inloggen of registreren. Wanneer je wel een gebruikersaccount hebt op de website, kun je naast de standaardfunctionaliteit ook reacties en productreviews plaatsen. Onder het diagram staat per zogenaamde usecase een uitwerking.



Use Cases

Naam	Inloggen op website
Samenvatting	De actor logt in op de website
Actor	Gast
Aanname	Actor bevindt zich op de website
Beschrijving	<ol style="list-style-type: none"> 1) Actor selecteert: "Inloggen" 2) Het systeem toont het inlogscherf 3) Actor vult de gevraagde gegevens in [1] 4) Het systeem stuurt de actor terug naar de hoofdpagina
Uitzondering	[1] Wanneer de combinatie van gebruikersnaam en wachtwoord onbekend is, wordt een foutmelding getoond.
Resultaat	De actor is nu ingelogd op de website

Naam	Registreren op website
Samenvatting	De actor registreert zich op de website
Actor	Gast
Aanname	Actor bevindt zich op de website
Beschrijving	<ol style="list-style-type: none"> 1) Actor selecteert: "Registren". 2) Het systeem toont het registratiescherf 3) Actor vult de gevraagde gegevens in [1] 4) Het systeem toont een melding dat het registeren gelukt is 5) Het systeem stuurt de actor door naar de loginpagina
Uitzondering	[1] Als een van de gevraagde gegevens niet geldig is, wordt een foutmelding getoond
Resultaat	De actor heeft een gebruikersaccount geregistreerd op de website.

Naam	Nieuwsartikelen opvragen
Samenvatting	De actor bekijkt het nieuws
Actor	Gast, Gebruiker
Aanname	Actor bevindt zich op de website
Beschrijving	<ol style="list-style-type: none"> 1) Op de hoofdpagina is het laatste nieuws te zien. Als de actor besluit meer nieuws te willen zien, volgt deze de volgende stappen: 2) In het navigatiemenu kiest de actor 'Nieuws' 3) Het systeem stuurt de actor door naar de nieuwspagina
Uitzondering	
Resultaat	Actor heeft een overzicht van al het nieuws

Naam	Reviews opvragen
Samenvatting	De actor bekijkt de reviews
Actor	Gast, Gebruiker
Aanname	Actor bevindt zich op de website
Beschrijving	<ol style="list-style-type: none"> 1) Op de hoofdpagina zijn de laatste 3 reviews te zien. Als de actor besluit meer reviews te willen zien, volgt deze de volgende stappen: 2) In het navigatiemenu kiest de actor 'Reviews' 3) Het systeem stuurt de actor door naar de reviewpagina
Uitzondering	
Resultaat	Actor heeft een overzicht van alle reviews

Naam	Zoeken van producten
Samenvatting	De actor zoekt een product op via de prijsvergelijker
Actor	Gast, Gebruiker
Aanname	Actor bevindt zich op de website
Beschrijving	<ol style="list-style-type: none"> 1) In het navigatiemenu selecteert de actor 'Pricewatch' 2) Het systeem stuurt de actor door naar de prijsvergelijker 3) Actor typt een productnaam of een deel daarvan in de zoekbalk 4) Actor bevestigt de zoekopdracht 5) Het systeem toont een lijst met alle product waarvan de naam overeenkomt met de zoekterm
Uitzondering	
Resultaat	Actor heeft een overzicht van producten

Naam	Productdetails opvragen
Samenvatting	De actor bekijkt details van een product
Actor	Gast, Gebruiker
Aanname	Actor bevindt zich op de prijsvergelijker-pagina
Beschrijving	<ol style="list-style-type: none"> 1) Actor zoekt een product zoals bij use case beschreven, of kiest een product uit de lijst 2) Het systeem toont de aanvullende gegevens van het product
Uitzondering	
Resultaat	Actor heeft een overzicht van productdetails

Naam	Productreview plaatsen
Samenvatting	De actor schrijft een productreview
Actor	Gebruikers
Aanname	Actor bevindt zich op de prijsvergelijker-pagina
Beschrijving	<ol style="list-style-type: none"> 1) Actor zoekt een product zoals bij use case beschreven, of kiest een product uit de lijst 2) Actor bekijkt de details van het product zoals beschreven in de use case 3) Actor selecteert 'Productreview schrijven' wanneer de actor het product heeft gevonden waar hij/zij een review voor wil plaatsen. 4) Het systeem stuurt de actor door naar de review-pagina 5) De actor beoordeelt het product op een aantal punten en schrijft eventueel een stuk tekst [1] 6) Het systeem verwerkt de beoordeling
Uitzondering	[1] Voor een review is minimaal een algemene beoordeling verplicht. Wanneer deze niet geselecteerd is wordt een foutmelding getoond.
Resultaat	De beoordeling wordt zichtbaar bij de productdetails

Naam	Reactie plaatsen
Samenvatting	De actor plaatst een reactie bij een item
Actor	Gebruiker
Aanname	Actor bevindt zich op pagina van een item (review, nieuws)
Beschrijving	<ol style="list-style-type: none"> 1) Actor navigeert selecteert 'reactie plaatsen' onder aan het artikel 2) Het systeem past de interface zodanig aan de een reactie getypt kan worden 3) De actor typt een reactie en bevestigt met 'versturen' [1] 4) Het systeem voegt de reactie toe aan het item
Uitzondering	[1] De reactie moet minimaal 2 karakters lang zijn, als dit niet het geval is, wordt de reactie niet verstuurd en wordt er een foutmelding getoond.
Resultaat	Actor is uitgeschreven uit het systeem.

Acceptatie Test Plan

Inleiding

In het acceptatie test plan, kortweg ATP, stel ik vast welke tests de applicatie moet doorstaan om als geslaagd te kunnen worden gezien. Met een testmatrix toon ik aan welke functionaliteit de verschillende testen dekken. Aan het einde van het ATP staan een conclusie gebaseerd op de testcriteria, uitgevoerde testcases en eventueel toelichting bij gefaalde testcases.

Alle testen worden uitgevoerd met **Visual Studio 2015 op Windows 10**. Het project wordt uitgevoerd met de project instellingen op **Platform Target 64-bit** en **64-bits IIS debugger**.

Acceptatiecriteria

De applicatie wordt als geslaagd beschouwd wanneer minstens **100,00%** van alle testen succesvol uitgevoerd zijn. Dit percentage is vastgesteld op basis van de volgende feiten:

- Alle testcases behandelen functies van de hoogste prioriteit
- Het aantal testcases is beperkt
- Optionele functionaliteit word niet getest

Vooralsnog worden alleen de belangrijkste functies getest. In het geval dat de applicatie uitgebreid is met extra functionaliteiten, zal het aantal testcases ook toenemen om deze functies te testen.

Testmatrix

In de testmatrix is de dekking van de requirements door de het acceptatie test plan aangegeven. De matrix en het test plan horen dus bij elkaar. Aanvankelijk worden alleen must-functionaliteit meegenomen in het testplan. Wanneer hieraan voldaan wordt komen de should- en ten slotte de could-functionaliteiten.

Testmatrix Individuele opdracht: Tweakers .net

Funcie / Testcase	T_LOGIN	T_REGISTER	T_NEWS	T_REVIEWS	T_PSEARCH	T_PDETAILS	T_PREVIEW	T_COMMENT
Inloggen	x							
Registreren		x						
Nieuws Bekijken			x					
Review Bekijken				x				
Product Zoeken					x			
Productdetails Bekijken						x		
Review Plaatsen							x	
Reactie Plaatsen								x

Acceptatie Test plan

In het acceptatie test plan staan de stappen die een voor een genomen moeten worden om de requirements te testen. De stappen moet duidelijk genoeg uitwerkt zijn zodat iemand die geen ervaring heeft met het programma, de testen eenvoudig kan uitvoeren. De applicatie is geslaagd als 87,5% van de tests met succes uitgevoerd kunnen worden.

Testcase identificatie	Beschrijving van uit te voeren stappen	Verwachte resultaat	Geobserveerde resultaat	Resultaat van testcase
T_LOGIN	1. Gebruiker selecteert 'Inloggen' 2. In het nieuwe venster vult de gebruiker als gebruikersnaam 'egestas@nulla.ca' in en als wachtwoord "NOM22YBP3JC". Beide waarden zijn hoofdlettergevoelig. 3. Gebruiker bevestigt het inloggen	Gebruiker is succesvol ingelogd en wordt doorgestuurd naar de hoofdpagina.	Doorgestuurd naar hoofdpagina. Rechtsboven in het scherm verschijnt de naam 'LynnzOr'	GOED
T_REGISTER	1. Gebruiker selecteert 'Registeren' 2. In het nieuwe venster vult de gebruiker als Email testmail@mail.com , als gebruikersnaam "Testgebruiker02" in en als wachtwoord "Testwachtwoord". 3. Gebruiker bevestigt het registeren	Gebruiker heeft succesvol een account aangemaakt en wordt doorgestuurd naar de login pagina.	Doorgestuurd naar loginpagina, het e-mailadres wat zojuist is opgegeven is al ingevuld bij het veld 'gebruikersnaam'	GOED
T_NEWS	1. Gebruiker selecteert 'Nieuws'	Het systeem toont een chronologisch aflopende lijst met de laatste nieuwsitems	Nieuwsartikels worden getoond.	GOED
T_REVIEWS	1. Gebruiker selecteert 'Reviews'	Het systeem toont een chronologisch aflopende lijst met de laatste reviews	Reviews worden getoond.	GOED

T_PSEARCH	1. Gebruiker selecteert 'Pricewatch' 2. In het nieuwe venster voert de gebruiker "Samsung" in de zoekbalk in. 3. De gebruiker bevestigt de zoekopdracht'	Het systeem toont een lijst van alle producten van het merk Panasonic	Een televisie van Samsung wordt getoond.	GOED
T_PDETAILS	1. Gebruiker volgt stappen van T_PSEARCH 2. Gebruiker klikt op een willekeurig product uit de lijst	Het systeem toont de details van het aangeklikte Samsung-product	De prijzen en specificaties van de televisie worden getoond.	GOED
T_PREVIEW	1. Gebruiker volgt de stappen van T_PDETAILS 2. Gebruiker selecteert 'Review schrijven' 3. In het nieuwe venster waardeert de gebruiker het product met een 4. Als tekst voert de gebruiker "Testreview test 123" in bij samenvatting. Het veld Review word leeg gelaten. 4. Gebruiker bevestigt de review 5. Gebruiker bekijkt de reviews door 'Gebruikersreviews' te selecteren	Het systeem stuurt de gebruiker terug naar de pagina met productdetails. Gebruiker stelt vast dat de review bij het product staat.	Teruggestuurd naar productdetails. Na het drukken op de knop Gebruikersreviews doorgestuurd. Daar stond de nieuwe review staat bovenaan.	GOED
T_COMMENT	1. Gebruiker selecteert 'Nieuws'. 2. Gebruiker selecteert een willekeurig nieuwsartikel uit de lijst van het nieuwe venster. 3. Gebruiker typt een reactie in het menu onder het artikel. 4. Gebruiker bevestigt zijn reactie	Het systeem stuurt de gebruiker opnieuw naar de pagina van het betreffende artikel. Gebruiker stelt vast dat zijn reactie is toegevoegd onder aan het artikel.	Reactie onderaan toegevoegd	GOED

Conclusie

Aan de hand van de gestelde eisen in de acceptatiecriteria is het programma geslaagd.

Ontwerp

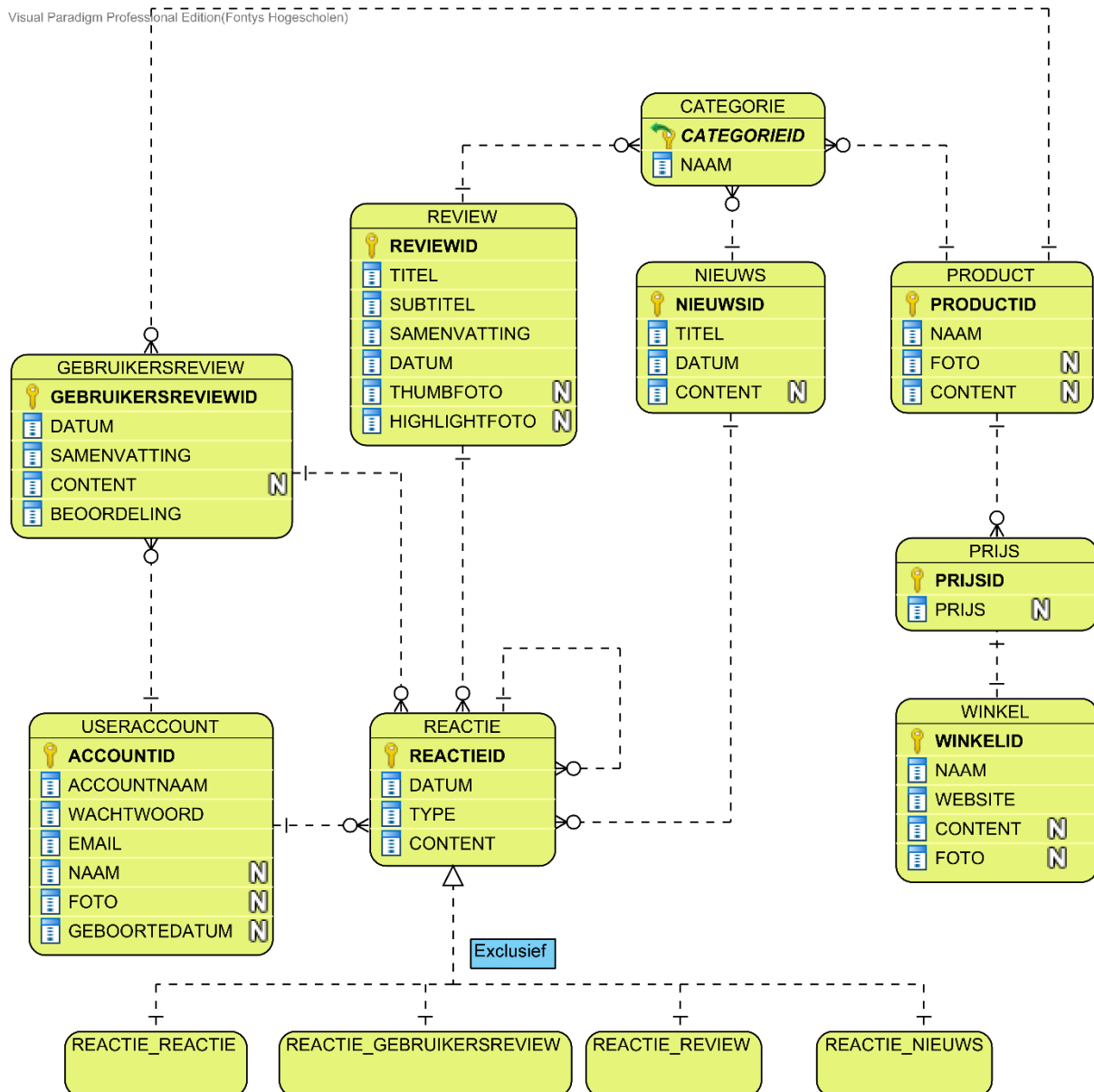
Inleiding

Het ontwerpen van de applicatie is begonnen met het herzien van de opgeleverde database uit DB2. Het Entiteit-Relatie Diagram en Database Ontwerp stromen dan ook voort uit die opdracht. Dit maakt de weg vrij om snel te beginnen aan het domeinmodel. Na het domeinmodel wordt ook nog kort de infrastructuur behandeld in dit deel van het document.

Om het verschil tussen planning en realiteit in beeld te brengen, heb ik het origineel geplande klassendiagram laten staan onder het kopje Domeinmodel. Het nieuwe en actuele klassendiagram volgt daarna.

Entiteit-Relatie Diagram

Visual Paradigm Professional Edition(Fontys Hogescholen)

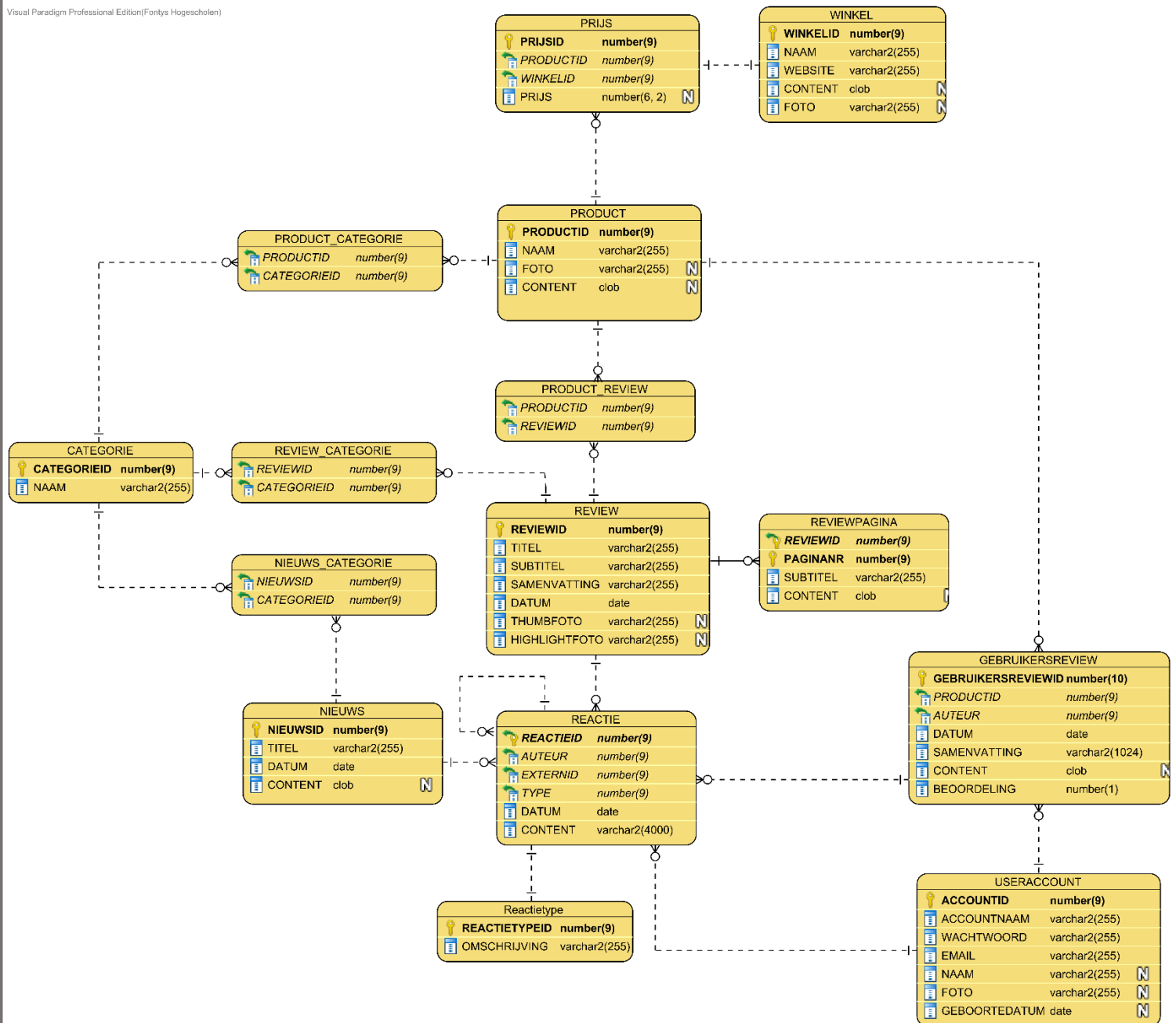


ERD-toelichting

Op basis van de case beschrijving uit de ontwikkelopdracht van DB2, heb ik het volgende ERD opgesteld. Deze dient als basis van het database ontwerp wat hierop verder gaat. Attributen waar een N achter staat mogen leeggelaten worden. Recursieve relatie en subtypering zijn beide aanwezig rondom de reactie-tabel. Objecten die centraal staan in het ERD zijn nieuws, reviews, reacties en reviews.

Database Ontwerp

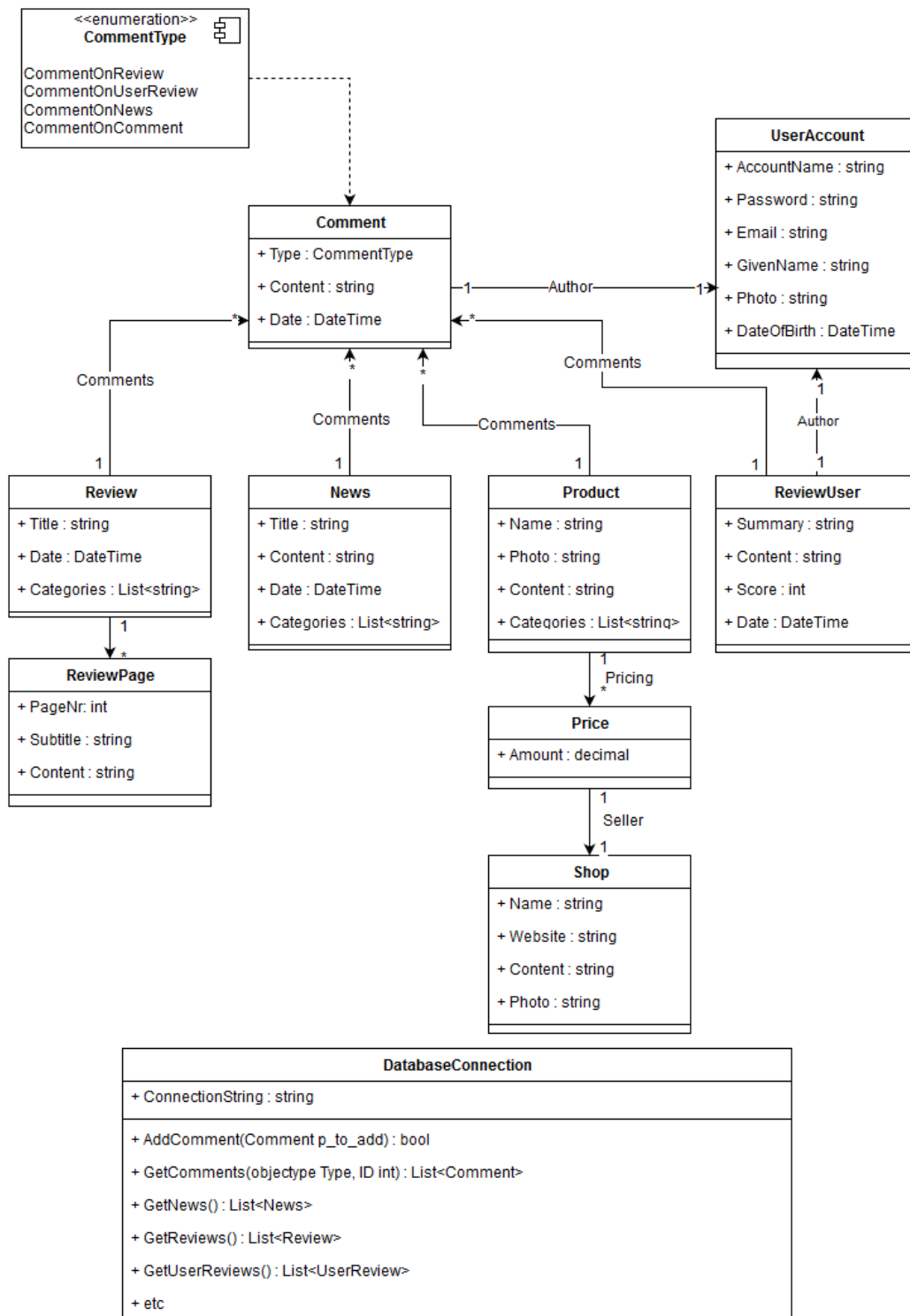
Visual Paradigm Professional Edition(Fontys Hogescholen)



DBO-toelichting

Het DBO is een doorontwikkeling van het ERD en toont de database zoals deze uitgewerkt is. Ook dit ontwerp komt voort uit de ontwikkelopdracht van DB2. Tabellen die zichtbaar worden zijn de koppeltabellen tussen categorie en de diverse anderen entiteiten. Daarnaast is er ook een tabel in het leven geroepen om de verschillende pagina's van de reviews op te slaan. De subtypering uit het ERD heeft geleid tot het ontstaan van de entiteit Reactietype.

Domeinmodel



Domeinmodel Toelichting

Het domeinmodel is een vroege versie van het klassendiagram. Te zien is dat de reacties op verschillende objecten gelijk mee is genomen als attribuut. Naast dit, en de verschillende database-entiteiten, is er een enumeratie die als attribuut van Comment aangeeft waarop de comment gericht is.

Klassendiagram



Klassendiagram Toelichting

Het klassendiagram is uiteindelijk een stuk anders gewonder qua verbindingen dan aanvankelijke gepland. Alle klassen uit het domeinmodel hebben het uiteindelijke klassendiagram wel gehaald.

Veel entiteiten zijn los komen te staan door de manier waarop ik de reacties heb aangepakt, waardoor het geven van een attribuut bestaande uit nieuws, review of gebruikersreview al snel een beperking werd. Op veel pagina's had ik *de lijst* met reacties niet nodig, slechts het *aantal reacties*. Hierom is erbij news, review en userreview het attribuut CommentCount bijgekomen, wat het verlies van de lijsten opvangt. Op deze manier hoef ik voor veel pagina's niet twee query's te sturen naar de database, maar slechts één.

Ook bij het toevoegen van reacties leverde dit geen problemen op want door de enumeratie is het plaatsen van reacties voor nieuws, reviews of gebruikersreviews geen extra informatie nodig behalve het veranderen van het reactietype. Bovendien was het 'externe' ID van de reactie al bekend omdat deze veelal op de pagina geladen was.

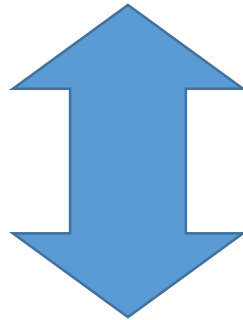
Verder zijn links onder de eerste van de klassen te zien die ik gebruikt heb voor refactoren. Helaas heb ik deze niet door kunnen voeren naar alle klassen vanwege de deadline.

Infrastructuur

De applicatie die ik ontwikkel maakt gebruik van een Oracle-database. In principe maakt het niet uit of dat deze online (bijvoorbeeld Athena) staat of lokaal, maar voor het argument stabiliteit heb ik gekozen voor een lokale database. Hierom heb ik het script om het database aan te maken, aangevuld met extra testdata en is deze ook aangeleverd bij de applicatie.



ASP.NET



ORACLE®

Om de communiceren met de database gebruik ik een referentie naar *Oracle.DataAccess V2.112.2.0* welke geïnstalleerd kwam met de 11g versie van de database die ik gebruik. Omdat deze library niet draait binnen 32-bit IIS is het belangrijk dat de IIS op 64-bit wordt gezet binnen Visual studio. Dit kan gedaan worden door de volgende optie aan te vinken:

In Visual Studio > Tool > Options... > Projects and Solutions > Web Projects

Vink **USE THE 64 BIT VERSION OF IIS EXPRESS FOR WEB SITES AND PROJECTS** aan.

Omdat er nu gebruik gemaakt wordt van een 64-bit IIS is het bovendien belangrijk om het doelplatform van Visual Studio op 64-bit te zetten. Dit kan gedaan worden via de project instellingen.