**Casus Fotoproducent**

**Themasemester Enterprise Software**

**Netwerk Open Hogeschool ICT**

**Concept versie 1.1**

**Michael Franssen (**[**michael.franssen@fontys.nl**](mailto:michael.franssen@fontys.nl)**)**

In deze casus kijken we naar een bedrijf dat foto's print voor schoolfotografen en portretfotografen bij grote winkelketens (niet de reguliere vakantiekiekjes van de particulier). Deze producent levert hoge kwaliteit en bedrukt ook speciale producten zoals muismatten, tassen, T-shirts, mokken, canvas, dibond (soort metaalplaat) etc. De fotografen leveren de foto's in hoge resolutie aan op DvD. Alle foto's worden afgedrukt en krijgen een uniek nummer. De schoolfoto's worden via de schoolfotograaf aan de school bezorgd en dan aan de scholieren. De portretfoto's worden aan de winkelketens geleverd, waar de klanten ze komen ophalen. De betaling van deze eerste fotoseries gebeurt bij ontvangst en volgt dezelfde route terug naar de fotoproducent.

**Huidige proces voor nabestellingen:**

Voor het nabestellen van foto's belt de klant nu naar de fotoproducent en geeft dan de gewenste bestelling door. De foto's kunnen worden verdeeld in individuele foto's (de portretfoto's van een gezin of de individuele schoolfoto) en groepsfoto's (de klassenfoto's van de schoolfotograaf). Individuele foto's kunnen alleen door de geportretteerde worden nabesteld en groepsfoto's in principe door iedereen. De klant krijgt telefonisch een bestelnummer en totaalbedrag en moet dat van te voren storten op de bankrekening van de fotoproducent. De storting wordt handmatig opgezocht en dan wordt de nabestelling geproduceerd en verzonden. Deze manier van nabestellen is arbeidsintensief, foutgevoelig en onhandig.

**Verbeterwens:**

De fotoproducent wil het nabestellen automatiseren. Het idee is dat alle klanten een inlogcode krijgen waarmee de foto's en bijbehorende producten via een website kunnen worden nabesteld. Het beoogde proces is dan als volgt: de fotograaf levert de foto’s bij de producent aan. Deze produceert de initiële serie als vanouds, maar plaatst de foto’s tevens op de website. Per serie foto’s wordt dan een account aangemaakt waarmee die serie kan worden nabesteld. De foto’s op de website zijn in een lage-resolutie (om webserver diskruimte te sparen en misbruik te voorkomen; klanten moeten niet zelf de foto's gaan uitprinten). Binnen het bedrijf staan de hoge resolutie foto's opgeslagen op een grote storage. Een kaartje met de bijbehorende inlogcode wordt door het systeem geproduceerd en samen met de initiële serie geleverd aan de klant. Na het inloggen kunnen klanten extra exemplaren van hun eigen foto's en producten online bestellen met daarop een gekozen foto.

De website moet previews laten zien en de te bestellen producten plaatsen in een winkelwagen. De inhoud van de winkelwagen kan via een online kassa (zoals Paypal, Google-checkout, Ogone of iDeal) worden betaald en is dan besteld. Betaalde producten worden bij de producent in hoge resolutie klaargemaakt om af te drukken. Bovendien wordt er een indexkaart klaargemaakt met daarop een kleine afbeelding van alle producten binnen de bestelling en het adres van de klant.

De prijzen van de producten kunnen worden ingesteld op de website. Ten behoeve van de financiële administratie moeten ook overzichten kunnen worden gemaakt van geplaatste en betaalde bestellingen.

**Eisen (“constraints”):**

* Het product moet in C# geschreven zijn.
* Het product voldoet aan de user interface eisen van de opdrachtgever.
* Het product moet gedistribueerd zijn - klanten en fotoproducent zijn op verschillende locaties.

**Opdrachten:**

1. Ontwerp een architectuur waarmee het bovenstaande gerealiseerd kan worden. In die architectuur dienen de tools te zijn opgenomen voor het publiceren van de foto’s op de website, de website zelf, het betaalsysteem en de tool voor het produceren van de hoge resolutie foto’s voor het afdrukken.
2. Werk voor de verschillende onderdelen van de architectuur een meer gedetailleerd ontwerp uit. Maak daarbij gebruik van design-patterns. Leg ook vast hoe de verschillende onderdelen van het systeem met elkaar communiceren. Houd bij het ontwerp rekening met de volgende aspecten:

* Gebruikers mogen niet in staat zijn om foto’s van een andere klant te bestellen.
* Gebruikers mogen niet in staat zijn om de database van de website te manipuleren op ongeoorloofde wijze (bijvoorbeeld: zelf bestellingen als betaald aan te merken of prijzen te veranderen)
* Gebruikers mogen niet in staat zijn om foto’s van een andere klant te bekijken.
* De service moet stabiel zijn. Het publiceren van een serie foto’s kan langere tijd in beslag nemen, dus moet de applicatie bestand zijn tegen het uitvallen van de publiceerapplicatie en/of de website tijdens het publiceren. Bij het hervatten na een crash is het niet acceptabel als alles opnieuw moet worden gepubliceerd.
* Portabiliteit: de webservice moet goed werken op verschillende platformen, zoals Windows, OSX, IOS en Android. De publiceerapplicatie mag zich beperken tot het platform van de klant.
* Schaalbaarheid: je ontwerp moet bestand zijn tegen onverwacht succes. Bij sterke groei van de verkoopactiviteiten moet het systeem goed blijven werken.
* Usability: het betreft een bestelsite voor individuele klanten. Het systeem moet zich dus zelf wijzen en bestellen moet zo eenvoudig mogelijk kunnen gebeuren.
* Geef aan welke constraints aan de data worden gewaarborgd in de database en welke in de software (en waarom).

1. Maak van het ontwerp een Software Architectuur Document (SAD) zoals gebruikelijk in het Rational Unified Process (RUP) (zie ook <http://www.cs.toronto.edu/~wl/teach/407/2002/rup-sad.html>).  Een voorbeeld hiervan staat ook op de ELO.
2. Implementeer in concept minimaal de website, het betaalsysteem PayPal[[1]](#footnote-1) en de applicatie voor het publiceren van de foto’s. Maak tijdens het implementeren gebruik van unit-tests om je applicatie te testen en om richting klant de correctheid zoveel mogelijk te waarborgen.

1. Iedereen met een e-mailadres kan op PayPal een betaalaccount aanmaken en daar documentatie downloaden die uitlegt hoe volledig automatisch PayPal betalingen op een website kunnen worden geaccepteerd. [↑](#footnote-ref-1)