## Aanpak / Ontwerp / Uitwerking

Planning gemaakt daar aangehouden, systematische aanpak

(planning/ functioneel ontwerp met UML bv. screenshots van het (toekomstige) product plus beschrijving/ eventueel iets over de testrapporten (testrapport te groot denk aan een bijlage)

Na het gesprek hebben we een analyse gemaakt (v1) en een schets van hoe onze toekomstige website eruit gaat komen te zien. De functies die de website moet hebben zijn:

…

…

…

..

Verder hebben wij twee testrapporten gemaakt. In het eerste rapport hebben we goed duidelijk aangegeven wat er nog moest gebeuren (wat nog niet goed ging) en wat voor priotiteit en einddatum hieraan zit gekoppeld. Tijdens het maken van onze tweede testrapport konden we alles op ‘dicht’ zetten.

In de loop van het project hebben we nog een aantal aanpassingen doorgevoerd in onze analyse. Zie ….

Versies analyse

Problemen en oplossingen

wij alle 3 geen kennis hadden van de werking van ASP.net was dit in het begin nog erg zoeken maar door het bekijken en uitwerken van tutorials zijn we er goed in geslaagd om kennis op te doen over ASP.net. Nadat we wat kennis in huis hadden zijn we begonnen met het bouwen van de website. Tijdens de bouw zijn we tegen een aantal problemen aangelopen. De problemen die de meeste tijd hebben gekost zijn:

* SQL queries
* Het gebruik maken van een SMTP server
* Editor in contact formulier
* Error handling
* Watermerk over de foto’s
* PayPal sandbox
* Mailen van de bestelde foto’s

**SQL queries**

Het maken van de SQL queries heeft ons redelijk veel tijd gekost, dit kwam vooral doordat wij gebruik maken van meerdere tabellen die aan elkaar zijn gekoppeld met foreign keys. Er is een koppeling tussen producten en categorieën en een koppeling tussen gebruikers en rollen. Door deze koppeling was het wat lastiger om de juiste waarden op het scherm te tonen.

**Het gebruik maken van een SMTP server**  
Aangezien wij op meerdere manieren gebruik maken van de mail functionaliteit hadden wij een SMTP (mail server) nodig zodat wij konden testen of dit mailen goed gaat. Uiteindelijk kwamen wij erachter dat google hiervoor een service aanbied waar wij dan nu ook gebruik van maken. Wij hebben een gmail account aangemaakt en met deze gegevens maakt de website nu gebruik van een smtp server van google waarover al het mail verkeer wordt geregeld.

**Editor in contact formulier**

Wij wouden de bezoeker van de website graag de mogelijkheid geven om op een gemakkelijke manier de opmaak van het bericht in het contactformulier aan te passen. Na een poosje op internet te hebben gezocht hebben we dit uiteindelijk opgelost met Ajax Control Toolkit, in deze Toolkit zit een HTML editor welke wij hebben geïntegreerd in het contactformulier.

**Error handling**

Als er dingen fout gaan op een website heb je liever niet dat deze dan compleet onderuitgaat maar dat hij de gebruiker een keurige foutmelding geeft. We zijn er uiteindelijk achtergekomen dat we dit kunnen opvangen in de web.config file. Hierin hebben we een stukje code toegevoegd die alle zoals het in ASP.net genoemd customerrors opvangt en dan doorstuurt naar een error page.

**Watermerk over de foto’s**

Wij kwamen er redelijk snel achter dat die op twee manieren kon worden opgelost, via de DrawString en DrawImage functie. Uiteindelijk hebben wij voor DrawImage gekozen. Het is nu zo dat wanneer er een foto wordt geupload de website automatisch ook een preview versie van deze foto maakt en hier 4 maal een plaatje met de tekst ‘© Milanov’ bovenop tekent.

**PayPal sandbox**

Wij hebben tijdens één van onze gesprekken met de klant afgesproken dat het betalen via PayPal zou verlopen. Dit wouden wij natuurlijk ook graag testen zonder met echt geld te betalen. Dit is ons uiteindelijk gelukt doormiddel van de paypal sandbox. Dit is speciaal voor ontwikkelaars ontwikkeld om te testen of paypal goed functioneert. Aangezien dit nog in BETA is, is het op dit moment alleen mogelijk om het met dollars te testen.

**Mailen van de bestelde foto’s**

We waren er redelijk snel achter hoe dit moest maar ons probleem was dat we maar 1 foto per keer als bijlage konden meesturen. Dit hebben we opgelost door met een loop door alle bestelde foto’s heen te lopen en deze 1 voor 1 als bijlage in de mail toe te voegen voordat deze wordt verzonden.

Reflectie Yme