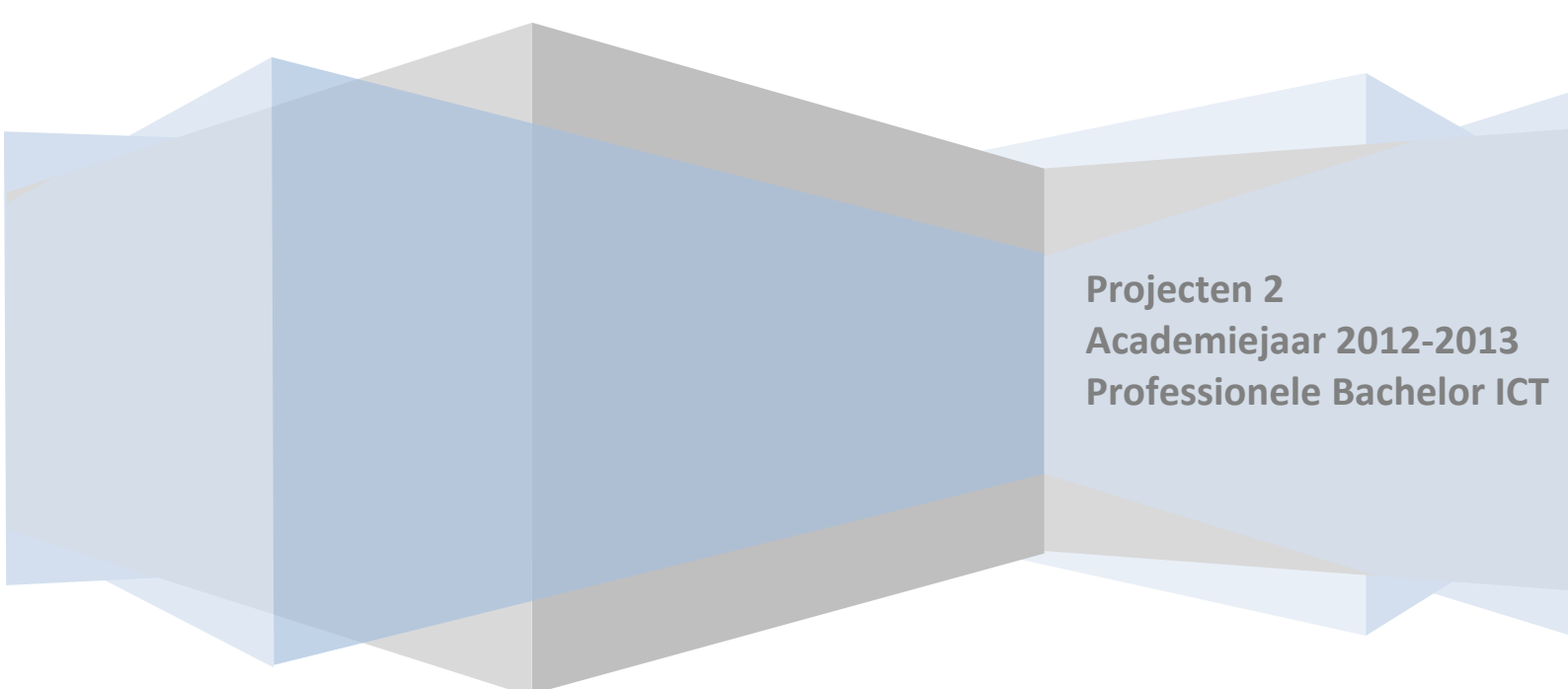


Modulair cpanel

Brecht Bonte

Niels Timmermans

Cedric Van Bockhaven



Projecten 2
Academiejaar 2012-2013
Professionele Bachelor ICT

Inhoud

Vastleggen projectvoorstel	2
Omschrijving van het originele idee	2
Uitwerken van uiteindelijk idee	2
Uiteindelijk projectonderwerp	3
Omschrijving van het bedoelde eindresultaat van het project.....	3
Doelpubliek & Context	3
Innovatief aspect van het project.....	3
Beschrijving en functionaliteit van je eindresultaat.....	3
Beschrijving van gebruikte technologieën, platform	4
Planning	5
Milestones	5
Werkplanning	6
Rapportering.....	6

Vastleggen projectvoorstel

Omschrijving van het originele idee

De bedoeling is om een modulair controlepaneel te maken waarop een gebruiker de verschillende services (Apache, MySQL, FTP, iptables, ...) op een linux-distro kan beheren.

Storingspunten bij huidige oplossingen zijn onder andere de trage verwerking van opdrachten, en de leercurve ervan (denk maar aan Virtualmin's opsplitsing van PHP onder Webmin, terwijl Apache onder Virtualmin valt).

We denken er aan om Node.JS te gebruiken waarvan de efficiënte opmerkelijk hoger ligt tegenover de traditionele Perl-scripts of Apache/PHP cpanels.

Dankzij Node.JS's asynchrone geaardheid zou deze al een stap voor hebben op traditionele synchrone/gemultiplexte systemen.

Uitwerken van uiteindelijk idee

Bij de opstartfase van het project zijn we op school enkele malen samengekomen om de verschillende ideeën samen te leggen.

We hadden enkele ideeën omtrent 3D-applicaties, zoals een simulatie van hoe je kapsel eruit gaat zien. Het Toledo-voorstel op de projectensite werd in overweging genomen, maar leek ons geen echte uitdaging.

Het cpanel-idee leek ons het beste omdat het een uitdaging geeft: Node.JS biedt een relatief jonge ontwikkelomgeving die, hoewel ze nog in haar kinderschoenen staat, al veel bekendheid heeft verworven door haar efficiëntie.

Uiteindelijk projectonderwerp

Omschrijving van het bedoelde eindresultaat van het project

Een performante linux-applicatie in Node.JS die de gebruiker toelaat op eenvoudige manier enkele services te beheren. Volgende services stellen we voorop: FTP (vsftpd/proftpd), MySQL, en bijkomend Apache. De gebruiker krijgt een webinterface voorgeschoteld waarop hij de verschillende modules kan beheren.

Doelpubliek & Context

Het controlepaneel is bedoeld om op een gebruiksvriendelijke en laagdrempelige manier de gebruikers toegang te geven tot het beheer van verschillende services. Het kan bedoeld zijn voor systeembeheerders die hun eerste stappen zetten in een linux-omgeving, of op termijn voor eindgebruikers van een hostingprovider.

Innovatief aspect van het project

Dankzij het asynchrone aspect van Node.JS is het in staat om een hoger aantal tasks tegelijkertijd te verwerken zonder dat de gebruiker hier last van ondervindt. Er is maar een kleine barrière tussen client- en servercode, waardoor het developmentproces sneller verloopt, en aanpassingen makkelijker kunnen gebeuren. Omdat javascript zowel op client- als servervlak kan gebruikt worden, is code die oorspronkelijk enkel als serverside-code bedoeld was, ook in de browser gebruikt worden.

Dankzij het modulaire aspect van hoe we onze applicatie willen uitbouwen, zouden gebruikers of derde partijen makkelijk hun eigen modules kunnen toevoegen.

Beschrijving en functionaliteit van je eindresultaat

- Node.JS server applicatie met modules:
 - Minimaal:
 - FTP (ProFTPd/vsftpd)
 - Database (MySQL)
 - Optioneel:
 - Webserver (Apache, PHP)
 - Mail (postfix)
 - Firewall (iptables)

Beschrijving van gebruikte technologieën, platform

- Platform
 - Linux (Debian) server
- Technologieën
 - App
 - HTML
 - Bootstrap => HTML/CSS framework
 - Javascript
 - (Coffeescript)
 - Libraries
 - Node.JS => Javascript interpreter
 - EJS / Jade / Vash => Template engine
 - ExpressJS => Web app framework (MVC)
 - SequelizeJS => Database ORM
- Technieken
 - Bash-scripting
 - Config-file parsing
- Communicatie
 - SVN
 - Redmine => Bugtracker

Planning

Milestones

- Opzetten ontwikkelomgeving
 - Node.js server installeren
 - Synchronisatie met git/svn, integratie met Cloud9
- Interface (Herfstvakantie)
 - Keuze van core libraries, boilerplate
 - Boilerplate aankleden
 - Modulariteit inwerken
 - Asynchroniteit: “task manager” maken/testen
- FTP-module
 - Parsing config files
 - Gebruikers aanmaken
 - Directory-mapping / jailed directories
 - Permissions
- Databank-module (Polyvalente week)
 - Parsing config files / query creation
 - Gebruikers + Databases aanmaken
 - Mapping user <-> DB
 - User, Db, Host Permissions
- Webserver-module (minimal) (Kerstvakantie)
 - Vhost creation (also port/SSL specific)
 - (Apache Module management, CGI tuning: setuid, max procs, logging, ...)
 - Automatic DB / FTP creation
 - PHP configuration
 - Integrity checks of manually modified files
- Optional: DNS (minimal)
 - Modification of records
- Optional: Mail (minimal)
 - User management
 - Forwarding / Direct modification of MX-servers in DNS
- Optional: Firewall (minimal)
 - Creation of ACCEPT/DROP on specific ports
- Optional: agent: turn the whole thing into a client-server system

Werkplanning


3	Concretiseren idee
4	Opzoekingswerk
5	Development omgeving opzetten

Herfstvakantie

Webinterface

Polyvalente week

FTP, MySQL

6		Coderen, Bugfixes
7		
8		
9		
10		
11		

Kerstvakantie

Webserver

Kerstvakantie

Mail, Firewall

12	Release
----	---------

Rapportering

- Face-to-face (14-daags)
- Redmine (roadmap / issue reporting)
- SVN