Software Requirements Specifications voor Schedule-Generator

Matthias Caenepeel Adam Cooman Alexander De Cock
Zjef Van de Poel

23 februari 2011 Versie 0.1

Aanpassingsgeschiedenis

. 23/2/2011 versie 0.1: Aanmaak document

Inhoudsopgave

1 Introduction

1.1 Purpose

Het doel van de SRS is om een overzicht te geven van alle functionaliteiten die er moeten voorzien worden.
Het doelpubliek is het team dat aan het project werkt en de docent die het team begeleidt en evalueert.

1.2 Scope

Schedule generator: deze software (geschreven in Java) zal in staat zijn om een lessenrooster samen te stellen dat aan bepaalde voorwaarden voldoet.
Website: dit is de software die de site omvat waarop de gebruikers zullen werken.
Database: deze zal de informatie over het lessenrooster bijhouden en ordenen.
Servelets: deze software zal ervoor zorgen dat de site doet wat de gebruiker vraagt.

1.3 references

IEEE Recommended Practice for Software Requirements Specifications De organisation pdf die de opdracht bevat.

1.4 Overview

De rest van de SRS zal de software vereisten verder uitdiepen.

2 Overall description

2.1 Product functions

Gebruikers krijgen een gepersonaliseerde account waarop zij hun eigen lessenrooster alsook dat van andere richtingen kunnen opvragen (en in het geval van sommige gebruikers kunnen aanpassen).
Het programma bevat een database waarin informatie over de verschillende vakken wordt bijgehouden (wie is de docent, hoeveel studenten zijn er ingeschreven, waar wordt het gedoceerd,...)
Het bevat ook een schedule generator die in staat is een lessenrooster te genereren die aan bepaalde voorwaarden voldoet.

2.2 User characteristics

De gebruikers zijn studenten die in staat om zijn om de verschillende lessenroosters te bekijken en docenten die in staat zijn het rooster op te vragen, maar evenwel aanpassingen te maken aan de informatie van de vakken waarvoor zij verantwoordelijk zijn.

2.3 constraints

Het geheel moeten worden gedraaid op de server Wilma die als besturingssysteem Linux heeft.
Alle code moet open source zijn.
Er moet geprogrammeerd worden in Java.
Men heeft internet nodig om aan de diensten te kunnen.

3 Specific requirements

3.1 External interface requirements

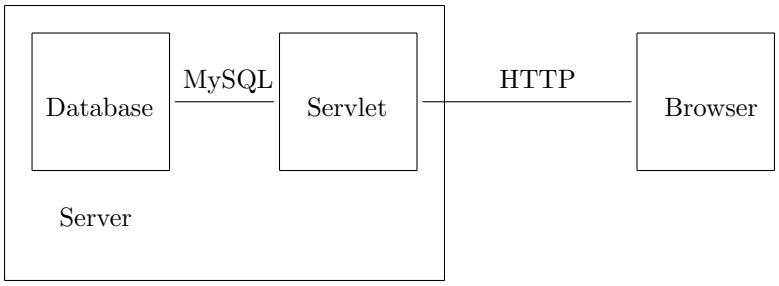
3.1.1 User interfaces

De opdracht gebiedt ons om met een website als user interface te werken. Om dit te verwezenlijken zal XHTML in combinatie met CSS gebruikt worden (mogelijk ook javascript). Na het inloggen krijgt de gebruiker een pagina te zien met verschillende tabs. Voor elke gebruikersklasse en de functionaliteiten die ze krijgen bestaat een ander tabblad. Op die manier is het gemakkelijker om functionaliteiten toe te voegen en overzichtelijk weer te geven. Guests krijgen bijvoorbeeld maar een enkele tab te zien waarin ze een naam kunnen intypen en waarin de kalender weergegeven wordt. Studenten krijgen een tweede tab erbij waarin ze hun vakkenlijst kunnen aanpassen etc.

3.1.2 Communications and software interfaces

Om de communicatie tussen de browser van de gebruiker en de server te doen wordt gebruik gemaakt van HTTP. De HTTP paketten zullen op de server geïnterpreteerd worden door servlets die op de server uitgevoerd worden via Tomcat. De servlets zijn in JAVA geschreven en genereren de gevraagde XHTML en CSS code die dat terug naar de gebruiker gestuurd worden.

De servlets communiceren met de database via MySQL en interpreteren de gegevens voor de gebruiker. Het proces kan overzichtelijk weergegeven worden in volgend blokschema:



Overzicht van de communicatie en de gebruikte protocols tussen de verschillende delen van het programma

De verschillende versies software die we gebruiken zijn de volgende:

. MySQL: JDBC driver for MySQL 5.1.15

. Tomcat

3.1.3 Hardware interfaces

De Hardware interfaces die we gebruiken worden vastgelegd door de opdrachtgever. Ons programma moet op Wilma kunnen draaien. De hardware interface die we hebben is dus een Linux besturingssysteem.

3.2 Functional requirements

ALEXANDER

3.2.1 User class 0: Unknown

login
enter as guest
Link naar email van beheerder

3.2.2 User class 1: Guest

Bekijk rooster van student, educator, lokaal
Bekijk eigenschappen van Vakken
Bekijk eigenschappen van Studenten
Bekijk eigenschappen van Proffen
Bekijk eigenschappen van Lokalen
Link naar email van beheerder

3.2.3 User class 2: Student

Breidt de functionaliteit van guest uit (kan kijken naar rooster)
Vakkenlijst aanpassen (door program toe te voegen of enkele vakken)
kan gegevens aanpasse (login naam, paswoord, ...)

3.2.4 User class 3: Educator

Breidt ook guest uit (kan kijken naar rooster)
Vakken aanpassen (enkel die die hij zelf geeft), constraints, description,...
Beschikbaarheid aanpassen

3.2.5 User class 4: Facility Manager

Kan Lokalen aanpassen
Kan Lokalen toevoegen/verwijderen
...

3.2.6 User class 5: Program Manager

Kan proffen op vakken zetten
Kan vakken per programma aanpassen
Kan vakken toevoegen

3.2.7 User class 6: Account Manager

Kan accounts wijzigen/toevoegen/verwijderen,... van Student en Educator

3.2.8 User class 7: Secretaresse

Heeft elke functionaliteit van guest, Program Manager, Account Manager, Facility Manager
Kan Rooster aanpassen

3.2.9 User class X: Admin

Kan alles
genereren
Kan Managers maken

3.2.10 Opmerking over User Classes

We verkiezen een methode waarbij de rechten van een account niet volledig vast liggen door zijn user class, maar veel losser kunnen bepaald worden. De user classes uit de vorige puntjes zijn templates, richtwaarden waar van afgeweken kan worden. De structuur moet dus per feature beschreven worden in een latere versie van het SRS.

ZJEF

3.3 Performance requirements

Schatting maken van aantal gebruikers dat tegelijkertijd aanwezig kan zijn daaruit volgt de grootte van de codedatabase en snelheid.

3.4 Design constraints

Linux, Wilma, Open-source

3.5 Software system attributes

3.5.1 Security

Als de gebruiker zich aanmeldt, wordt een code gegenereerd voor die gebruiker (random nummer) die lokaal bijgehouden wordt door de gebruiker.
Bij elk commando van de gebruiker wordt de code meegegeven.
Op de server wordt een tijdelijke lijst bijgehouden die de code aan accounts verbindt. (en inlogtijd,... om ze na een tijd weg te kunnen smijten als de gebruiker niet letterlijk uitlogt).
Bij elk commando van de gebruiker wordt de code in de lijst opgezocht, de rechten van de bijhorende gebruiker gecontroleerd en al dan niet kan het commando uitgevoerd worden.

Logboek van aanpassingen door Managers wordt bijgehouden.