

# Software Requirements Specifications voor Schedule-Generator

Matthias Caenepeel      Adam Cooman      Alexander De Cock  
Zjef Van de Poel

23 februari 2011 Versie 1.1

## Aanpassingsgeschiedenis

- . 23/2/2011 versie 0.1: Aanmaak document, toevoeging
- . 25/2/2011 versie 0.2: Toevoeging tekst vanaf Performance Requirements tot einde
- . 28/2/2011 versie 1.0: Toevoeging functionele vereisten, verbeteringen doorgevoerd
- . 17/3/2011 versie 1.1: Opmerkingen van opdrachtgever in acht genomen en aanpassingen doorgevoerd waar nodig

## Nog aan te passen

- . De inhoud van secties 3.1.1 en 3.1.2 horen eigenlijk thuis in een SDD. Alles in heel sectie 3 moet geformuleerd worden in termen van requirements, dus met een vaste structuur, een unieke ID, pre- en postcondities etc. Een requirement moet zo geformuleerd worden dat ze als het ware kan afgevinkt worden op het einde.
- . 3.2.12 "Roosterplanner" lijkt ons een heel belangrijk en non-triviaal onderdeel van de opgave. Het lijkt ons hier aangewezen iets gedetailleerdere (en eventueel opgesplitste) requirements te voorzien. Bvb I.5 "Melden van conflicten in het rooster" zou toch specifieker kunnen. Tot hoeveel detail wil je minimaal voorzien, gaat hij exact aan kunnen geven waar en waarom een conflict zich voordoet (soms kan dit complex zijn).

# Inhoudsopgave

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Scope . . . . .	4
1.3	Overview . . . . .	4
<b>2</b>	<b>Overall description</b>	<b>5</b>
2.1	Product functions . . . . .	5
2.2	User characteristics . . . . .	5
2.3	constraints . . . . .	5
<b>3</b>	<b>Specific requirements</b>	<b>6</b>
3.1	External interface requirements . . . . .	6
3.1.1	User interfaces . . . . .	6
3.1.2	Communications and software interfaces . . . . .	6
3.1.3	Hardware interfaces . . . . .	7
3.2	Functionele Vereisten . . . . .	8
3.2.1	Categorie op basis van prioriteit . . . . .	8
3.2.2	Categorie op basis van datapad . . . . .	8
3.2.3	Gebruikertype . . . . .	8
3.2.4	Identificatie . . . . .	10
3.2.5	Opvragen van gegevens . . . . .	10
3.2.6	Beheren van vakken . . . . .	11
3.2.7	Beheren van programma . . . . .	12
3.2.8	Beheren van faculteiten . . . . .	13
3.2.9	Beheren van faciliteiten . . . . .	13
3.2.10	Beheren van accounts . . . . .	14
3.2.11	Beheren van beperkingen . . . . .	15
3.2.12	Roosterplanner . . . . .	16
3.2.13	Beheren van lessenroosters . . . . .	17
3.2.14	Beheren van logboek . . . . .	17
3.2.15	Overige . . . . .	18
3.2.16	Opmerking over User Classes . . . . .	18
3.3	Performance requirements . . . . .	18
3.4	Design constraints . . . . .	18
3.4.1	Security . . . . .	18

# 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 Overview

De rest van de SRS zal de software vereisten van de schedule generator verder uitdiepen. De verschillende gebruikers zullen gedefinieerd worden en de functionaliteiten die die gebruikers krijgen zullen opgesomd en besproken worden.

## **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 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. Daarnaast zijn er ook verschillende klassen beheerders die verantwoordelijk zijn voor groepen vakken, groepen studenten, gebouwen en het lessenrooster zelf. Deze beheerders moeten de nodige aanpassingen ook op de website kunnen doorvoeren.

### **2.3 constraints**

De grootste constraints werden door de opdrachtgever vastgelegd. De eigenlijke opdracht staat in het SPMP, maar de constraints die erin stonden waren de volgende:

- Alle code moet open source zijn.
- Er moet geprogrammeerd worden in een object-georiënteerde programmeertaal. Er werd gekozen voor JAVA, omdat het team daar het meest ervaring mee heeft.
- Alle input van gebruikers moet via de website gebeuren.
- De website en het programma draaien op de server Wilma aan de VUB.

## 3 Specific requirements

### 3.1 External interface requirements

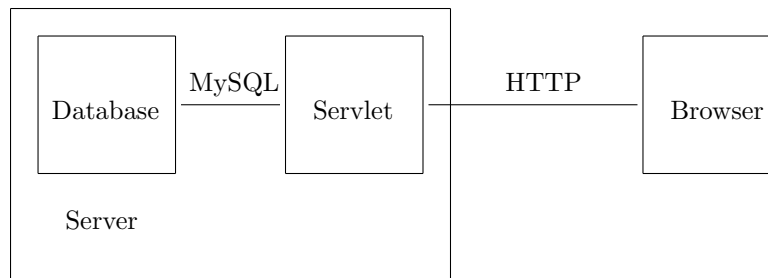
#### 3.1.1 User interfaces

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 pakketten 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 Functionele Vereisten

In deze subsectie van het SRS wordt een overzicht gegeven van alle functionaliteiten van de software. Om de ontwikkeling van de functionaliteiten tijdens het productie te kunnen opvolgen, wordt elke functionaliteit voorzien van een naam, code en omschrijving. Daarnaast worden de functionaliteiten ook nog eens opgedeeld in verschillende categorien en indien mogelijk toegekend aan een bepaald gebruikertype.

### 3.2.1 Categorie op basis van prioriteit

Enerzijds kan men de functionaliteiten opdelen volgens hun prioriteit. Dit leidt tot onderstaande categorien:

- *noodzakelijke functionaliteiten*: functionaliteiten waarvan de aanwezigheid in het eindproduct wordt gegarandeerd omdat ze noodzakelijk zijn voor een minimale werking van het systeem. Deze functionaliteiten hebben de hoogste prioriteit tijdens de verwezenlijking van het project.
- *mogelijke functionaliteiten*: functionaliteiten waarvan de uitvoering haalbaar is maar zonder garantie op aanwezigheid in het eindproduct. Deze functionaliteiten hebben een matige prioriteit
- *extra functionaliteiten*: functionaliteiten waarvoor de prioriteit laag is.

### 3.2.2 Categorie op basis van datapad

Anderzijds kan men functionaliteiten ook opdelen volgens het pad die de data tijdens hun uitvoering volgt. Dit geeft aanleiding tot volgende categorien

- *Invoerfunctionaliteiten*: alle functionaliteiten die de gebruiker toelaten informatie op te sturen naar de database op de server.
- *Uitvoerfunctionaliteiten*: alle functionaliteiten die de gebruiker toelaten informatie op te vragen uit de database op de server.
- *Verwerkingsfunctionaliteiten*: alle functionaliteiten die gegevens, zonder tussenkomst van de gebruiker, uit de database verwerken en de daarbij bekomen resultaten terug schrijven naar de database op de server.

### 3.2.3 Gebruikertype

Om bepaalde functionaliteiten te kunnen plaatsen in een context wordt het gebruikertype dat toegang heeft tot deze functionaliteit indien mogelijk vermeld. Vandaar volgt hier een korte beschrijving van de gebruikertypes:



- *Software beheerder*: Verantwoordelijke voor het beheren technische aspecten van de software. Deze omvatten o.a. installeren van de software, laten aanmaken van de databasestructuur en het configureren van de lessenroosterplanner. Daarnaast heeft deze gebruiker ook toegang tot een logboek en kan hij bepaalde gebeurtenissen ongedaan te maken.
- *Account beheerder*: Deze gebruiker kan de omschrijving van bestaande gebruikertypes wijzigen en nieuwe gebruikertypes definieren. Hij is ook verantwoordelijk voor het aanmaken van de andere beheerder accounts.
- *Rooster beheerder*: Deze gebruiker is instaat lessenrooster manueel aan te passen, lessenroosters door de roosterplanner te laten genereren en algemene beperkingen voor de roosterplanner in te stellen (bvb: feestdagen). De rooster beheerder zal ook eventuele conflict situaties in het rooster moeten oplossen.
- *Student beheerder*: Deze gebruiker is verantwoordelijk voor het aanmaken van de studenten accounts en het toekennen van programmas en vakken aan studenten. De vakken toegewezen door de student beheerder zullen in rekening worden gebracht tijdens het opstellen van de lessenroosters.
- *Docent beheerder*: Deze gebruiker is verantwoordelijk voor het aanmaken van de docent accounts en het toekennen van vakken aan docenten.
- *Programma beheerder*: Deze gebruiker is verantwoordelijk voor het aanmaken van vakken, programmas en faculteiten.
- *Faciliteit beheerder*: Deze gebruiker is verantwoordelijk voor het aanmaken van de faciliteiten. Deze faciliteiten zullen vervolgens door roosterplanner worden verdeeld over de verschillende vakken in overeenkomst met de beperkingen opgelegd door de docent van het vak.
- *Docent*: Gebruiker waarvan een lijst van gegeven vakken wordt bijgehouden. De docent is verantwoordelijk voor het beheer van de vakken die hem door een programma beheerder zijn toebedeeld. De docent is ook instaat beperkingen op te leggen die in rekening moeten worden gebracht tijdens het opstellen van de lessenroosters.
- *Student*: Gebruiker waarvan een lijst van gevolgde programmas en vakken wordt bijgehouden. Deze lijst wordt aan de student toegekend door een student beheerder maar kan door de student zelf worden aangevuld. Deze aanvullingen worden echter niet in rekening gebracht tijdens het opstellen van de lessenroosters. Een student kan, naast de gegevens waarover een gast beschikt, ook nog een persoonlijk lessenrooster opvragen dat wordt opgesteld aan de hand van zijn vakkenlijst.
- *Gast*: Gebruiker waarvan geen gebruikersgegevens worden bijgehouden in de database op de server. Deze gebruiker heeft enkel toegang tot de geplande lessenroosters. Deze kan hij opvragen op basis van vak, student, programma, docent en semester.

- *Onbekende gebruiker*: Gebruiker die zich nog niet heeft gedentificeerd tegenover het systeem. Deze gebruiker heeft enkel toegang tot de aanmeldpagina van de website waarbij hij kan kiezen zich aan te melden of de rest van de site te betreden als gast.

### 3.2.4 Identificatie

#### A.1 Aanmelden

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *onbekende gebruiker*

Omschrijving: Doorsturen van gebruikersnaam en wachtwoord om toegang te krijgen tot de account gebonden gegevens en functionaliteiten

#### A.3 Aanmelden als gast

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *onbekende gebruiker*

Omschrijving: De website betreden zonder een account. Zie gast voor meer details.

#### A.3 Afmelden

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: De site verlaten

### 3.2.5 Opvragen van gegevens

#### B.1 Opvragen van faculteit

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: Laat toe na te gaan welke programmas zijn gebonden aan welke faculteit

#### B.2 Opvragen van programma

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: Laat toe na te gaan welke vakken behoren tot welk programma

#### B.3 Opvragen van vak

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: Geeft toegang tot vak gebonden gegevens

#### B.4 Opvragen van student

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: Laat toe de gevolgde programmas en vakken van een student na te gaan

B.5 Opvragen van docent

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *docent*

Omschrijving: Laat toe na te gaan voor welke vakken een docent verantwoordelijk is

B.6 Opvragen van lessenrooster op programma niveau

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: Geeft een rooster weer met alle vakken van een programma

B.7 Opvragen van lessenrooster op vak niveau

Categorie: *mogelijke uitvoerfunctionaliteit*

Gebruikertype: *iedereen behalve onbekende gebruiker*

Omschrijving: Geeft weer wanneer een vak is gepland

B.8 Opvragen van lessenrooster op student niveau

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *student*

Omschrijving: Geeft het persoonlijk rooster van een student weer

B.9 Opvragen van lessenrooster op docent niveau

Categorie: *noodzakelijke uitvoerfunctionaliteit*

Gebruikertype: *docent*

Omschrijving: Geeft het persoonlijk rooster van een docent weer

### 3.2.6 Beheren van vakken

C.1 Vakken aanmaken

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Programma beheerder*

Omschrijving:

C.2 Vakken verwijderen

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Programma beheerder*

Omschrijving:

C.3 Vakken wijzigen als docent

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Docent*

Omschrijving: Laat toe de vakomschrijving te wijzigen

- C.4 Vakken wijzigen als beheerder  
Categorie: *noodzakelijke invoerfunctionaliteit*  
Gebruikertype: *Programma beheerder*  
Omschrijving: Laat toe de naam, docent en omschrijving van een vak te wijzigen
- C.5 Vakken koppelen aan een student  
Categorie: *noodzakelijke invoerfunctionaliteit*  
Gebruikertype: *Student beheerder*  
Omschrijving: Zie student beheerder
- C.6 Vakken koppelen aan een docent  
Categorie: *noodzakelijke invoerfunctionaliteit*  
Gebruikertype: *Docent beheerder*  
Omschrijving: Zie docent en docent beheerder
- C.7 Vakken onderverdelen in programmas  
Categorie: *noodzakelijke invoerfunctionaliteit*  
Gebruikertype: *Programma beheerder*  
Omschrijving:
- C.8 Vakken inladen uit bestand  
Categorie: *extra invoerfunctionaliteit*  
Gebruikertype: *Programma beheerder*  
Omschrijving: Via een nader te bepalen bestand type vakken inladen in de database om gegevens uit ander database makkelijk te kunnen importeren.

### **3.2.7 Beheren van programma**

- D.1 Programmas aanmaken  
Categorie: *noodzakelijke invoerfunctionaliteit*  
Gebruikertype: *Programma beheerder*  
Omschrijving: Programmas worden gebruikt om vakken te bundelen
- D.2 Programmas inladen uit bestand  
Categorie: *extra invoerfunctionaliteit*  
Gebruikertype: *Programma beheerder*  
Omschrijving: Via en nader te bepalen bestand type programmas inladen in de database om gegevens uit ander database makkelijk te kunnen importeren.
- D.3 Programmas verwijderen  
Categorie: *noodzakelijke invoerfunctionaliteit*  
Gebruikertype: *Programma beheerder*  
Omschrijving:

- D.4 Programmas wijzigen  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder*  
 Omschrijving:
- D.5 Programmas koppelen aan een student  
 Categorie: *noodzakelijke/mogelijk invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder, student*  
 Omschrijving: Zie Programma beheerder
- D.6 Programmas onderverdelen per faculteit  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder*  
 Omschrijving:

### 3.2.8 Beheren van faculteiten

- E.1 Faculteiten aanmaken  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder*  
 Omschrijving: Faculteiten worden gebruikt om programmas in te bundelen
- E.2 Faculteiten inladen uit bestand  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder*  
 Omschrijving: Via een nader te bepalen bestand type faculteiten inladen in de database om gegevens uit ander database makkelijk te kunnen importeren.
- E.3 Faculteiten verwijderen  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder*  
 Omschrijving: Gebruiker kan kiezen of programmas en vakken van een faculteit mee worden verwijderd
- E.4 Faculteiten wijzigen  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Programma beheerder*  
 Omschrijving: Bepalen welke programmas tot welke faculteit behoren

### 3.2.9 Beheren van faciliteiten

- F.1 Faciliteiten aanmaken  
 Categorie: *mogelijke invoerfunctionaliteit*  
 Gebruikertype: *Faciliteiten beheerder*

Omschrijving: Faciliteiten omvatten zowel locaties (vb lokaal) als benodigdheden (vb projector)

F.2 Faciliteiten inladen uit bestand

Categorie: *extra invoerfunctionaliteit*

Gebruikertype: *Faciliteiten beheerder*

Omschrijving: Via een nader te bepalen bestand type faciliteiten inladen in de database om gegevens uit ander database makkelijk te kunnen importeren.

F.3 Faciliteiten verwijderen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Faciliteiten beheerder*

Omschrijving: Als een faciliteit wordt verwijderd moeten alle afhankelijke beperkingen ook verwijderd

F.4 Faciliteiten wijzigen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Faciliteiten beheerder*

Omschrijving:

### 3.2.10 Beheren van accounts

G.1 Gebruikers aanmaken

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving:

G.2 Gebruikers inladen uit bestand

Categorie: *extra invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving: Via een nader te bepalen bestand type gebruikers inladen in de database om gegevens uit ander database makkelijk te kunnen importeren.

G.3 Gebruikers verwijderen

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving: Alle gegevens van de gebruiker worden uit de database verwijderd

G.4 Gebruikers wijzigen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving:

#### G.5 Gebruikers blokkeren

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving: Een geblokkeerde gebruiker krijgt geen toegang tot het systeem tijdens het aanmelden

#### G.6 Gebruikertypes aanmaken

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving: Laat toe functionaliteit te bundelen in gebruikertypes op maat.

#### G.7 Gebruikertypes verwijderen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving: Sommige gebruikertypes zijn beschermd tegen verwijdering om verdere werking van de software te garanderen.

#### G.8 Gebruikertypes wijzigen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Account beheerder*

Omschrijving: Sommige gebruikertypes zijn beschermd tegen aanpassingen om verdere werking van de software te garanderen.

### 3.2.11 Beheren van beperkingen

#### H.1 Tijdsbeperking aanmaken

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder, docent*

Omschrijving: een tijdsbeperking legt op wanneer een vak wel of niet kan worden geplant

#### H.2 Tijdsbeperking verwijderen

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder, docent*

Omschrijving:

#### H.3 Tijdsbeperking wijzigen

Categorie: *noodzakelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder, docent*

Omschrijving:

#### H.4 Faciliteitbeperking aanmaken

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder, docent*

Omschrijving: een faciliteitbeperking legt op welke faciliteiten er nodig

zijn voor een vak

H.5 Faciliteitbeperking verwijderen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder, docent*

Omschrijving:

H.6 Faciliteitbeperking wijzigen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder, docent*

Omschrijving:

### 3.2.12 Roosterplanner

I.1 Configureren van roosterplanner

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Software beheerder, Rooster beheerder*

Omschrijving: Bepaald met welke gegevens de roosterplanner moet rekening houden

I.2 Starten van roosterplanner

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving: Data noodzakelijk voor de planning mag na het starten niet meer worden aangepast. Het starten van de roosterplanner bevriest dus sommige gegevens tot zijn planningstaak is gestopt.

I.3 Stoppen van roosterplanner

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving: Het handmatig stoppen van de roosterplanner. De planner kan alter weer worden gestart om zijn taak verder te zetten.

I.4 Status van de roosterplanner opvragen

Categorie: *mogelijke uitvoerfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving: Het handmatig opvragen van de status van de roosterplanner.

I.5 Melden van conflicten in het rooster

Categorie: *mogelijke verwerkingsfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving: Als de rooster planner geen rooster kan vinden dat voldoet meldt hij dit als een conflict



### 3.2.13 Beheren van lessenroosters

#### J.1 Lessenroosters bereken

Categorie: *mogelijke verwerkingsfunctionaliteit*

Gebruikertype: *Roosterbeheerder*

Omschrijving: De rooster planner berekend automatisch roosters die voldoen aan de opgegeven beperkingen.

#### J.2 Lessenroosters aanmaken

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving:

#### J.3 Lessenroosters verwijderen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving:

#### J.4 Lessenroosters wijzigen

Categorie: *mogelijke invoerfunctionaliteit*

Gebruikertype: *Rooster beheerder*

Omschrijving: Lessen roosters kunnen handmatig worden aangepast

### 3.2.14 Beheren van logboek

#### K.1 Logboek aanmaken

Categorie: *extra invoerfunctionaliteit*

Gebruikertype: *Software beheerder*

Omschrijving: Als een logboek is aangemaakt zullen belangrijke gebeurtenissen in het systeem worden bijgehouden

#### K.2 Logboek bekijken

Categorie: *extra uitvoerfunctionaliteit*

Gebruikertype: *Software beheerder*

Omschrijving:

#### K.3 Gebeurtenissen registreren

Categorie: *extra verwerkingsfunctionaliteit*

Gebruikertype: *Software beheerder*

Omschrijving:

#### K.4 Gebeurtenissen ongedaan maken

Categorie: *extra invoerfunctionaliteit*

Gebruikertype: *Software beheerder*

Omschrijving:

### 3.2.15 Overige

#### X.1 Aanmaken van de software databasestructuur via SQL

Categorie: *noodzakelijke verwerkingsfunctionaliteit*

Gebruikertype: *Software beheerder*

Omschrijving: De software brengt zelf een datastructuur aan die nodig is voor het opslaan van de gegevens

#### X.2 Aanpassen van de look en feel van de website

Categorie: *extra invoerfunctionaliteit*

Gebruikertype: *Software beheerder*

Omschrijving: Dit wordt mogelijk gemaakt door et toegankelijk maken van de opmaak bestanden.

### 3.2.16 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.

## 3.3 Performance requirements

Er zijn geen specifieke vereisten in verband met de snelheid van de software. Het is echter wel duidelijk uit de aard van het project, dat er een groot aantal gebruikers tegelijk de webinterface (en met gevolg de databases) moet kunnen consulteren.

## 3.4 Design constraints

De opdrachtgever heet de omgeving waarin de software moet draaien gespecificeerd.

De opgelegde beperkingen, met betrekking op het design, zijn de volgende:

- . Systeem moet draaien op een linux server, meer bepaald Wilma (<http://wilma.vub.ac.be/>)
- . Uitsluitend gebruik van open source software
- . User interactie gebeurt via een gebruiksvriendelijke grafische web interface met bovengenoemde server
- . Flexibiliteit van verschillende parameters instelbaar door de gebruiker

### 3.4.1 Security

#### User access restriction

Verskillende gebruikers krijgen verschillende rechten toegewezen. Deze bepalen tot welke informatie en tools hij toegang krijgt. Deze rechten zijn gekoppeld

aan de account van deze gebruiker. Na het inloggen zullen enkel de informatie en tools waarvoor de gebruiker gemachtigd is, getoond worden. Om verdere beveiliging te verzekeren, wordt ook de communicatie met de server beveiligd. Bij het inloggen krijgt de webinterface van de gebruiker een access code toegestuurd, op dat moment gegenereerd door de server. Deze houdt bij welke rechten bij deze code horen. De webinterface stuurt de verkregen code mee door met elke instructie naar de server, waarop deze kan controleren of de gebruiker gemachtigd is om de desbetreffende instructie uit te voeren.

### **Data integriteit**

Enerzijds zal er de mogelijkheid geleverd worden, aan de daarvoor gemachtigde gebruiker(s), om op de server een back-up van de databases (zoals accounts, leslokalen, vakken,...) te maken en desnoods een rollback uit te voeren. Anderzijds zal er op de server een logboek bijgehouden worden met de aanpassingen aan de databases, die door verschillende gebruikers gemaakt kunnen worden.

### **Account security**

Account paswoorden zullen, met een nog nader te bepalen, geëncrypteerd verstuurd en opgeslagen worden.