Software Requirements Specification

For

CURRICULUM SCHEDULER

Prepared By

Shaik Asif Ahmed

Y Himakar Sai Teja Reddy

P V Siva Krishna

Guide

M V Panduranga Rao

Date: 02.03.2014

INDIAN INSTITUTE OF TECHNOLOGY HYDERABAD

TABLE OF CONTENTS

Revision History

Author	Date	Changes Made	Version	Reviewed By

1. Abstract

The system to be developed is for scheduling all the courses semester wise in a department for a batch based on the input of a file which consists about course details and entering details regarding number of minimum and maximum credits that a student must complete every semester and details about each course. The scheduler should satisfy different constraints. It follows the IEEE standard for a requirements specification document, with some variations.

2. Introduction

2.1 Purpose

The purpose of this document is to outline the functional requirements for a course scheduling system semester wise for an academic department in a University. . This document is meant to explain the features of the course scheduler so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

2.2 Intended Audience and Scope

This document is the only one that describes the requirements of the system. It is meant for use by the developers or students, academic staff and will be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process.

More specifically, this system is designed as a drag drop system. A file which has information about the constraints and the courses is given as an input. It is processed in such a way that we should drag and drop the courses in the semester slot provided for them. An error message will be popped up if the selection of any course violates the constraints. Final output is exported into a file.

2.3 Definitions, Acronyms, Abbreviations

CSV: Comma separated value.
GUI: Graphical User Interface.
Course Type: Core course or Liberal Arts elective or Free elective or Department elective etc.

2.4 Developer's Responsibilities

The developer is responsible for (a) developing the system, (b) installing the software on the client's hardware, (c) minimum and maximum credits for each type of course each semester. (d) Provide individual details about each course present in the set of department courses which is given as an input in the form of a csv file.

2.5 Technologies used

Java Development Kit (**JDK 7**): It is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

Javax.swing: It is the primary java GUI widget toolkit. It is used in making GUI for the curriculum scheduler. It is part of Oracle's Java Foundation Classes (JFC) — an API for providing a graphical user interface (GUI) for Java programs.

OpenCsv-2.3.jar: Java library to handle read and write operations in files of CSV format.

2.6 Overview

Rest of the SRS will include 2 sections namely

Overall description:

This section gives an overview of the functionality of the product and some general constraints while making the software and some assumptions and dependencies that are assumed.

Specific requirements:

This section gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

3. Overall Description:

3.1 Product perspective

Course scheduler is intended to help the students, academic staff of a department in an institute to decide which courses to be put in for registration every semester from the set of department courses as per constraints department provide to them.

3.2 Product functions overview

In a department, there are a set of courses that a student should complete in all the eight semesters. Each course has a unique course ID. For each course there are some constraints. E.g., a course must be completed between particular semesters and prerequisites for a course etc.

The system is to produce that specifies a list of courses every semester for all the eight semesters. Courses are dragged and dropped in the semester slot given. If a course added

against the constraints specified, the system should produce a “conflict report” that the chosen course cannot be added to the particular semester slot and the constraint that is violated.

3.3 User characteristics

The main users of this system will be department secretaries, who are somewhat literate with computers and can use programs such as editors and text processors.

3.4 operating environment

No specific platform constraint as the software is written in java.

3.5 General assumptions and dependencies

a) Full working of Curriculum Scheduler is independent on any other external sources such as availability of Internet connection.

b) The prerequisite courses for any course have to be specified before that particular course in the input csv file.

c) Each course has a unique course ID.

4. Specific Requirements:

4.1 Inputs and Outputs

Input to the system is given in 2 steps semester details and course details and the system produces an output file of the final list of courses.

4.1.1 Input:

Step 1:

In the first step the user is expected to fill in the semester details. The user is asked to fill in the details as shown in the attachment below. User has to fill in the number of credits(min credits and max credits) of each course type that need to be completed in a given semester as per the details given by the academic section or the faculty of the department. Input is a number which is the number of credits(min and max).After submitting the details semester wise, we move on to step2.

Step 2:

In the second step, a file which contains course details is given as an input to the system. It is a csv file and is explained as follows.

Input file:

The input file contains the list of all the courses and their corresponding details like course ID, fractal (whether fractal or not),number of credits it carries, course type, prerequisite (if any), semesters between which it has to be completed (minsem and

maxsem) , elementary courses of which the course is a combination of if it is not a fractal course(if any).

The input file is a csv extended file which will be like as shown below.

CourseName	CourseID	Fractal	Credits	Type	PreRequisite	MinSem	MaxSem	Elementary Course1	Elementary Course2
Introduction to C and C++	CS2000	TRUE	3	CORE	NIL	1	2		
Introduction to Computer Networks	CS3001	TRUE	1	CORE	CS2000	4	5		
Detailed Study of Computer Networks	CS3002	TRUE	2	CORE	CS3001	5	6		
Computer Networks	CS3020	FALSE	3	CORE	CS2000	4	6	CS3001	CS3002

4.1.2 Output:

After all the processing, the output is exported into a csv file which shows final list of courses semester wise is shown below.

Output file:

Output file is a csv file which consists of following details:

Semester: Semester in which the courses are selected.
Course name: Name of the course which is a string.
Course ID: Unique course ID which will be of form CSxxxx etc.
Course Type.
Credits: Number of credits student will acquire if the course if completed.

4.2 Functional Requirements:

1. Determine the final course curriculum such that following constraints are satisfied:

(a) Number of credits for each type of course (core or LA etc.,) should not be less than min credits and should not be greater than max credits.

Min credits <= credits <= max credits.

(b) Number of credits of the elementary courses should sum up to the number of credits of the main course.

Credits(Elementary course 1 + Elementary course 2 +….) = credits (Parent course)

(c) A course must be scheduled at least in the minsem and at most in the maxsem.

Minsem <= semester <= maxsem.

(d) The course mentioned in the prerequisite (if any) for a particular course must be scheduled in one of the previous semesters than that of the course.

(e) The course and the corresponding lab must fall in the same semester.

(f) Total number of credits in the final course curriculum should sum up to 151.

After the input part is completed, we process the courses. The list of all the courses in the input file will be available to drag and drop in the semester slot given. We can drag and drop a particular course successfully only if the input details like minsem, maxsem, prerequisite, min credits and max credits etc., are satisfied.

A message box pops up in case an error occurs showing the constraint violated and course will not be added to the slot.

We describe the functional requirements by giving various use cases.

Use case 1: Filling Courses details with constraints.

Primary Actor: User
Pre-Condition: Input file should be in csv format.
Main Scenario:

1. User fills all the details of courses in a csv file.
2. A sample csv file is provided for user to fill details in required format.
3. Import file by initiating “Import” functionality in menu bar.
4. The course details are stored in corresponding data structures and displayed in the left pane of UI.

Use case 2: Filling Semester details with constraints.

Primary Actor: User
Main Scenario:

1. User initiates the “Semester Details” functionality in menu bar.
2. User fills individual semester details in the GUI.
3. User fills credits details for each type (core, free, liberal arts).
4. User submits after filling required details.

Use case 3: Processing Courses.

Primary Actor: User
Pre-Condition: All details of courses and semester should be filled.
Main Scenario:

1. User initiates the “Process Courses” functionality.
2. All courses and their elementary courses are displayed in a tree structure in upper part of left pane.
3. Eight semester tables are displayed in the lower part of left pane.
4. User drag and drop courses to the corresponding semester tables.
5. If any constraint is violated, a pop up is displayed showing the violated constraint. Here, the course is not dropped to the table.
6. After filling all courses, the user can export information in all tables to .csv file by initiating “Export” functionality.

4.3 External interface requirements:

User interface: The entire software is a stand-alone application in JAVA. The user screen is split vertically into two panes. The left pane contains the UI for filling semester credit constraints, processing courses, which expands and contracts as per user action. The right part displays the instructions related to operations that are specified on the left pane. Either one of the left pane or right pane can be expanded partially or completely.

4.4 Design constraints:

Software constraints: The system will run on all operating systems (Solaris, Linux, Mac OS X, WINDOWS versions) supporting JRE (Preferably latest versions). Since it is a written and developed entirely in java, it is platform independent.

Language requirements: The system's UI is to be built in plain English only.

Hardware constraints: The system will run on any workstations with 256 MB RAM or above, running supported OS specified above.

5. Acceptance Criteria

Before accepting the system, the developer must demonstrate that the system works on the course data for the last 8 semesters. The developer will have to show through test cases that all conditions are satisfied.