Y10@SWE

Software Requirements Specification

for ‘Student Grading Portal v 1.0’

Itar Mehak Arneja, Nishant Bansal, Rajat Vyas, Ravi Prakash

2013

THE LNMIIT JAIPUR

INDEX

Revision History

Name	Date	Reason for changes	Version
Group 16	05-02-2013	NIL	Version 1.0

1. Introduction

1.1 Purpose

This software package is the first version of “Student Grading Portal” which is being developed for The LNMIIT, Jaipur.

- To automate the process of Grading of students by the instructor.
- To provide a sophisticated GUI for students to view their academic performance.
- To enable instructors of a course to monitor the academic performance of students and advise them accordingly.
- To simplify marking the borders of various grade groups in a course via graphical methods.

1.2 Document Conventions

The following documentation conventions are followed in preparing this SRS:

a) All key-words related to the academics are formatted in *italics.*
b) SRS-Software Requirement Specification
c) OS-Operating System

1.3 Intended Audience and Reading Suggestions

This document is created for,

i) The Instructors of the course ‘Software Engineering’ for their review and monitoring progress of the project.
ii) The software development team for their use in analyzing the requirements.

1.4 Product Scope

The scope of the to-be-developed ‘Student Grading Portal’ software package is:

i) To cater to ALL types of *Courses* offered by the administration of LNMIIT to its students.
ii) To cater to the need of a suitable interface for all students and instructors of an offered course.
iii) To cater all activities of ‘Student Grading’ , viz, from the *Marking Scheme, Weightage* of *Quizzes, Attendace, Assignments, Projects, Midsem* and *Endsem* Examinations.

1.5 References

The following references are used in preparing this SRS:

i) The SRS Template and Sample sent by Prof. Ravi Gorthi.
ii) Wikipedia.

2. Overall Description

2.1 Product Perspective

The following diagram describes the high-level business process of the Grading Portal:

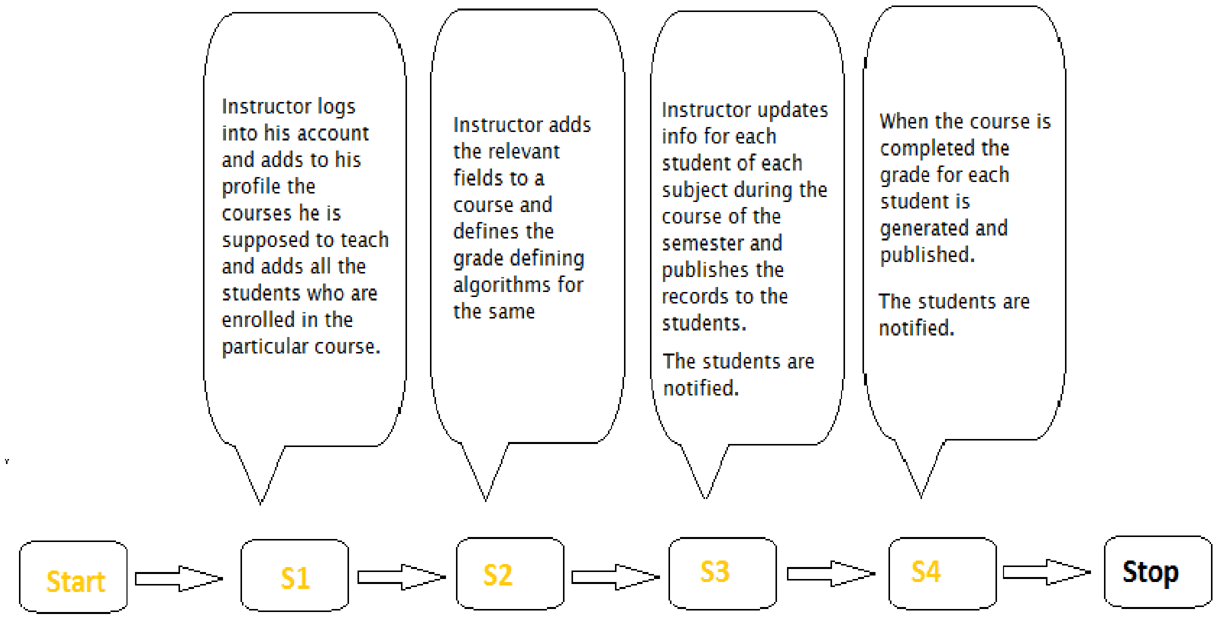
Instructor logs into his account and adds to his profile the courses he is supposed to teach and adds all the students who are enrolled in the particular course.
Instructor adds the relevant fields to a course and defines the grade defining algorithms for the same
Instructor updates info for each student of each subject during the course of the semester and publishes the records to the students.
The students are notified.
When the course is completed the grade for each student is generated and published.
The students are notified.
Start
S1
S2
S3
S4
Stop

Fig- Product Function Diagram

2.2 Product Functions

This software package is expected to offer the following services:

a) For the Course Instructors of LNMIIT:
   a. Facility to add and maintain the different courses offered by The LNMIIT curriculum; and a facility to define and maintain the process flow and grading fields of each course.

b) For the Students of the LNMIIT:
   a. Facility to go through their course progress and the percentiles and percentages of various fields in the course.

2.3 User Classes and Characteristics

This software package will be used by two categories of users:

a) **Course Instructors :** These users will add the courses that they are teaching, add the students that are enrolled in the course, fill in the marks and other details of individual students, define the algorithms that calculate the grades of a student and publish them.

b) **Students** : These users will use this software package to view their academic performances.

c) **Admin** : Handles discrepencies in the database and has the permissions to view all records of all students of all courses. Also, the Admin provides the instructor with the final list of all students enrolled in a course.

2.4 Operating Environment

This software package is expected to work in the following atmosphere:

i) Microsoft stack consisting of,

   a. OS - Windows 7, XP
   b. Eclipse IDE
   c. Java, PHP, JQuery, HTML, Java Applets
   d. MySQL for RDBMS on Apache server.

2.5 Design and Implementation Constraints

The design time constraints are:

i) The software package should be designed so as to handle the access by ~20 Instuctors 1 Admin and ~150 students concurrently.

2.6 User Documentation

i) This software package will come with a users manual as a guide to its interface actions.

ii) The details of the (i) Analysis, (ii) Design and (iii) Test Cases of this software package will be delivered along with this software.

2.7 Assumptions and Dependencies

The are no assumptions made.

3. External Interface Requirements

3.1 User Interfaces

The set of User Interfaces consists of,

i) To log in to their respective accounts by users.

ii) To Add and Modify courses, Add fields to a course such as Quizzes, Attendance, Midsem, Endsem and define their respective weightages in the course, save and publish the various data stored in the fields.

iii) To Read the ‘published’ data, by the students/Admin according to the permission of their respective accounts.

3.2 Software Interfaces

This software package should have an interface with,

i) Eclipse IDE on a Windows OS which acts as a platform for coding activities.

3.3 Communications Interfaces

This software package should be securely accessible through internet communication channels (wired or wireless).

4. System Features

The requirements of this software package are described per each category of User:

i) All requirements of LNMIIT’s Course instructors
ii) All requirements of LNMIIT’s students.
iii) All requirements of the portal’s Admin.

4.1 Use Case Activity for Instructor:

4.1.1 Functional Requirements:

1) The Instructor is provided with a unique User-id and Password (which (s)he can alter later) by the Admin.
2) The Instructor’s account consists of his Dashboard which in return contains links to other pages and pop-ups which assist the functionality of the module.
3) The functionality of the Instructors module contain : Logging in, Adding a new course,
   Updating info, Viewing course details, Populating The database through Excel sheets.

4.1.2 Stimulus-Response Table:

S.no	Stimulus from user	Response from module
1	“Log in”- Providing user-id and password and pressing login.	The System validates the login and redirects the user to his/her dashboard if successful, prints an error message “Retry” if not.
2	Clicking on “Add a new course” Button on the Dashboard.	Pops up a check-box list of all the available courses which the user can check, from where the user can “Add” and get redirected to dashboard with updated info.
3	On clicking “Course-info” on the Dashboard.	A list of all the added courses pops up and on clicking one of the links to a course the user is directed to the “Detail-page” of that course which shows in tabular from the last updated details of the course.
4	The user can “Manually edit the data table or “Upload” an existing Excel sheet to update the database.	On doing so and clicking “Update”, The “Detail-page” is reloaded with the updated information.

4.2 Use Case Activity for a Student:

4.2.1 Functional Requirements:

1) Every student is required to register himself up and create an account and then log in.
2) The student is required to fill in compulsory fields – Name, Roll no, Password, non-compulsory fields – Photograph, email-id.
3) The student Dashboard allows him/her to view and monitor accordingly his or her performance in a particular course.

4.2.2 Stimulus-Response Table:

S.no.	Stimulus from user	Response from module
1	Clicking “Register Yourself” on the home page.	A new window pops up which requires to be filled in some compulsory information(Roll no, First Name, Last name, Password) to successfully create an account. If the user leaves out a compulsory field the UI prompts him for it on “Register”.
2	Successfully “Register”-ing or on Log-in.	The user is redirected to his/her Dashboard where he can “Update info” of his account or view his coursewise details
3	“Update-info” button	A new window pops up where he/she can- Edit Name, Password, Photopgraph, Email-id and “Enroll” in a course.
4	“Enroll”	A dialog box containing the list of all offered *Courses* which can be ticked and submitted to get enrolled in single or multiple courses.
5	Clicking “Done” on the dialog box.	User Redirected to Dashboard with updated information.
6	“View course details” on dashboard.	Expands a list of courses he/she is enrolled in and clicking on one opens a new window where the details of the course last updated can be viewed in a tabular form.

4.3 Use Case Activity for Admin:

4.3.1 Functional Requirements:

1) Admin has Read/Write access to the database for the Account-level and Enrollment details but not the Academic records of a student.
2) Provides the Instructor the course wise enrollment list of students in an Excel sheet.
3) Required a central Dashboard to 'View' all details of all courses and students in a tabular form.

4.3.2 Stimulus-Response Table:

S.no.	Stimulus from user	Response from module
1	Logging in to Admin panel using admin id and password.	Redirects to a Dasboard which has various options to view the database in the form of required tables.
2	"Couse-wise list of Student enrollment"	Fetches the required tables from the database and Displays in tabular form.
3	"Student-wise list of enrollment"	Fetches the required tables from the database and Displays in tabular form.
4	"Student wise academic record"	Fetches the required tables from the database and Displays in tabular form.
5	"Course wise Academic record of Student"	Fetches the required tables from the database and Displays in tabular form.
6	"View course details" on dashboard.	Fetches the required tables from the database and Displays in tabular form.
7	"Print"	Generates a PDF from of the document being viewed and prints.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

This software should be able to handle the following load from the mentioned users:

1) Atleast 20 logged in instructors on an average of four hours a day, five days a week.

2) Atleast 300 students logged in to their accounts for 2 hours for a day of the week.

3) It should be able to handle a MySql -database corresponding to a strength of 50 instructors and 1000 students.

5.2 Safety Requirements

This software makes the process of student grading paperless however at the end of the semester a hardcopy of the Grade-Sheet has to be generated by the administration using the data forwarded by the course instructor and then given to each student.

5.3 Security Requirements

This software should,

i) Authenticate each user, who logs in;

ii) When the user performs any actions, Authorize him / her to perform the actions allowed for the user and display an error message if found to be not authorized.

5.4 Software Quality Attributes

- **Reliability:**

A failure should not occur while the users are making changes to their account. (Or while instructor is updating the marks and grades).

- **Extensibility/Maintainability:**

The software will be incremental type and any further modifications can be implemented depending upon the hardness level of modification.

- **Performance:**

All the transactions must be completed in less than 5 seconds.

- **Usability:**

This software is being offered to be used in a university scenario and the users of this module will be under graduate and post graduate students, instructors, the Dean, the Director, the registrar and the administrator.

- **Availability**:

All the hardware and the software related to this particular module come in the windows package. Care must be taken while installing the Windows environment on your machine because at the time of installation, all those packages should also be installed.

- **Security:**

The module must not be accessible by the outsiders (any person who is not a part of that institute) and hence, an authorization service is being implemented, so that only those students / instructors who are having a valid Password and Password will be given access to this module.

5.4 Business Rules

There are no specific business rules as such.

6. Other Requirements

Appendix A: Glossary

Minimal usage of abbreviations and shortcuts. The document is very easy to understand and the language is neat. Some of the abbreviations used are listed in Sec. 1.2.

Appendix B: To Be Determined List

Providing other features such as scaling up the s/w to show the grades of all eight semesters in the student account and notifications on the recent activity of any teacher.

*****