February 17, 2013

Integrated Campus

Software Requirement Specifications
Increment - I

Group 2

INTEGRATED CAMPUS | SEN-Winter 2013

Revision History:

Version	**Primary Author(s)**	**Description of Version**	**Reviewed By**	**Date Completed**
v_1.0	Ayush Jain, Vidhan Agarwal, Ishita Agrawal	This is the first version of SRS Increment - I	Vipul Garg, Pinky J Meena	17-02-2013

Approved By:

Prof. Vijay Kumar Chakka
Client, Integrated Campus

Contents:

Introduction

Purpose

The purpose of this document is to present a detail description of the requirements from the software under the project: “Integrated Campus”. The document presents all the specifications of the software - both the functional and the non-functional that would be implemented in the first increment of the product. This SRS aims to remove any ambiguity in the minds of the project team and the client too in terms of the functions of the product. The document also mentions the various constraints, assumptions and dependencies which the software suffers from.

The document is meant to be read by all the stakeholders of the project, primarily the client - **Prof. Vijay Kumar Chakka** so that all the requirements can be concretised and all the discrepancies regarding the features to be incorporated are removed.

Scope

With the exponentially growing dependency of the academic institutes on the internet, accessibility and manageability of the resources on both the student’s and the faculty's end is the need of the hour. Many a times user has to wander from one application to other for different kinds of information which is time consuming and it also increases the work load on the administration too which has to manage all these applications. Instead of carrying a separate basket for each egg, putting them all together in a single one seems to be a much better way out. And that is what the purpose of the project is. It is all about getting the important pieces of the academic picture on one canvas in an eye soothing way.
'INTEGRATED CAMPUS' as the name suggests, integrates all the essential requirements of an academic institutes from both the student's and the faculty's perspective on one platform whether it is academics related, interaction related, student attendance related, results related, so on and so forth.

Definitions, Acronyms, Abbreviations

- PDF - Portable Document Format
- PHP-PHP Hypertext Pre-processor
- RAM-Random Access Memory
- MySQL- My Structured query language
- XAMPP – X *(any of four different operating systems)*, Apache, MySQL, PHP and Perl
- PC -Personal Computer
- OS -Operating System

References

S.E.N, 2013, Group #2, Feasibility Report_v1.0

S.E.N, 2013, Group #2, Project Proposal_v1.0

S.E.N, 2013, Group #2, Project Plan_v1.0

S.E.N, 2013, Group #2, Survey

IEEE Guide to Software Requirement Specifications

Overview

After the introduction section, the document contains a section on the General Description of the product and another on the Specific Requirements. The General Description is expanded upon under the headings of Product Perspective, Product Functions, User Characteristics, General Constraints & Assumptions and Dependencies. The section on ‘specific requirements’ aims to provide an in-depth specification of the features that would be met out in the first increment. This section is described under the headings of functional requirements, non-functional requirements, formats of input and output and the design constraints.

General Description

Product Perspective

The software is totally self-contained, but will use the SMS API's. The software takes inspiration from Moodle- a popular course management feature that is already being used in the institute but many professors are still not comfortable using it due to its complex interface.

The software aims at implementing the features, which have been shortlisted and pruned on the basis of requirements of the client, the analysis of the surveys conducted in the student community at DA-IICT and the information obtained after personal interaction with many faculty members.

Product Functions

Attendance Feature:

The changes in the institute's academic policies have made attendance a mandate for all the students. This module is therefore of crucial importance to the institute and needs to be delivered in the very first increment itself as clearly specified by the client.

It is required that the TA's/Faculty allocated for any course be facilitated with a system wherein they can feed in the daily attendance records compiled manually by the TA's/Faculty himself in that particular course.

The system should be able to calculate the attendance percentage of each student with ease. If the attendance of any student starts to dip below a particular threshold (as mandated by the concerned professor) then the particular student is immediately brought into the notice of the professor and the professor can choose to contact the student directly or enable automated alerts to the student.

The professor can choose to look into attendance of all of his students, for not only their course but also for all/any of the other courses in which the students have registered so that general observations can be made about the student and the professor can get an estimate if the problem of low attendance is due to the students' lack of interest in his particular course or is it that the students have a general attitude of not attending the lectures.

The student can also check his attendance in each course on a daily basis. This would help him in planning his future leaves and minimising it if necessary.

As an additional feature, a SMS service has to be integrated into the module which can send a preformatted and automated text message to the student's parent's mobile devices if the attendance starts to dip below the threshold.

Polling Feature:

The polling feature aims at providing the faculty with a way to collect feedback from the students and analyse their grasp in the different portions/domains of the course. Such a feedback is highly necessary *within* the semester. Usually feedback is collected at the end of the semester when it is already too late to mend any impairment which the course might be suffering from or the students might be facing.

The results of the poll can be presented to the required professor whenever he wants either during the ongoing of a poll or after the poll is completed.

The professor can choose to implement any methodology to halt/terminate the polling process – either manually, or by prescribing a time limit.

File Upload Feature:

This features aims in providing the faculty the facility to upload the course material as and when the professor wants to disseminate the material. The students would be able to access these sources and download them to their own systems. A size restriction on the files to be uploaded would be implemented so that the server isn't burdened excessively.

The files that would be uploaded by the professor would be saved onto the server in a systematic style wherein each file gets saved onto the specific course subfolder under the folder that would be named as the professor's name itself.

Being an extremely important feature for both the faculty and the students (core feature), this functionality would be a part of the first increment itself.

User Characteristics

Student:

The student body would be the largest set of users of the software and also the more frequent one. They are expected to play a major part in improving the product through feedbacks once the first increment is delivered. All the students are technically sound and well - versed with the web. From the e-survey conducted in the student community it becomes clear that maximum students would want a web-based platform for better interaction with the professors, seniors and their batch-mates.

Any student would prefer a site that is easily navigable and takes less time to load. Students also have the tendency to judge the success of a website through its overall look and feel of a website. Thus this user-base is a large determiner of the way the software shapes up and the success of the software is subject to the approval by this community.

Faculty:

Faculties are very most important users of our software - many of the features are important to faculties for smooth completion of the course, of which they are the instructor. Features like attendance and polling will help them to get regular feedback to enable edition during the ongoing semester only instead of waiting till the end for the feedback.

This user base is quite learned and has a large vista of knowledge and experience. They have been using enough software's and surfing enough sites to determine if the software is workable for them within a few try-outs.

It is highly necessary that the professors find the software easy to use, appreciate simplicity in its interface and use it to meticulously plan their course and disseminate their course material.

Admin:

The administration is required to fix upon the courses that would be offered during a particular semester, maintain a registry of the professors available during the semester and map the different courses to the different professors. Assigning TA's will also be done by the admin only.

The admin would also be responsible for generating the lecture, lab and tutorial time table for the semester along with the examination schedules. The admin is also incharge of logistic issues like assigning different venues to different lectures being held at the same time.

System Personnel:

They are the most technically sound users of the website. They have full rights to the server and can fix the maximum load which the server can take. They would have the full control of the website along with the database and would be able to update the website or database whenever they want.

General Constraints:

This is the first time any of the team member has worked in a project of this scale and magnitude. It is a demanding call on the developers to learn the scripting languages involved before the actual start of the implementation phase.

Also we will be using third party SMS API, so for the alert messages we will depend on the provider of the API.

Safety and security of the website is an important concern as the server would contain sensitive and critical data. Any kind of tampering and hoodwinking could lead to serious repercussions for the client and the image of the developers.

It is necessary that the first increment be completed on time, so that the core modules as demanded by the client is thoroughly tested before the launch of the full product.

Assumptions and Constraints:

We assume that user should have internet connection to access the website, and he/she should be part of the institution to avail the facilities provided by the product. The user is assumed to have basic knowledge about websites, and should know how to navigate. The GUI is aimed to be user friendly so that person who have less or no experience, will also be able to understand and use the features provided by the product.

Specific Requirements

Functional Requirement:

System features:

1 Sign up

Description: The users will be automatically registered by the admin whenever they join the college.

Risk: Low

Effort: Would require a few person hours

Preconditions: The user should be either a registered student, faculty, or an employee (admin, system personnel) of the Institute.

2 Login

Description: User can use his/her username and password for login.

Risk: Low

Effort: Few Person Hours

Precondition: The user should be registered with a valid username and password.

Postcondition: User changes will be saved and can access the courses in which he has registered.

Input: Username and Password.

Output: User is notified about successful login by opening his profile page.

3 Forgot Password:

Description: User can get a system generated password if he/she has lost the password.

Risk: Low

Effort: Few Person Hours

Precondition: User should remember his username

Input: Username/Mail ID

Output: A new machine generated password is sent to user's preregistered email.

4 Change Password:

Description: User can change his password using this feature.

Risk: Low

Effort: Few Person Hours

Precondition: User should remember his username and password.

Input: Username and password

Output: Password is changed.

5 Creation of Courses:

Description: Admin can add courses which are offered in the semester.

Risk: Low

Effort: Few Person Hours

Precondition: Course should be offered in the semester, and some students must have enrolled for the course.

Input: Course and Instructor details

Output: New course is added

6 Student/Faculty/TA Assignment to the course:

Description: Assigning Students/Faculties/TA's to their respective courses.

Risk: Low

Effort: Few Person Hours

Precondition: Faculty should be the instructor, and student must have enrolled in the course.

Postcondition: They are now added to the course.

Input: Details of students who have enrolled and who have applied for TA ship.

7 Uploading File:

Description: Initially in first increment only faculties can upload files for course related material.

Risk: Low

Effort: Would require considerable Person Hours

Precondition: Faculty should login first.

Postcondition: The uploaded file can then be accessed from the courses page by students.

Input: File to be uploaded.

Output: Update is sent to all users (students who have enrolled in the course).

8 Adding Attendance:

Description: This will allow TA's or Faculties to add attendance data after every lecture. This will help them to keep track of the students who are attending the lectures and who are not.

Risk: Medium

Effort: Would require considerable Person Hours

Precondition: User should be a TA/Faculty of the course.

Postcondition: This will update the total percentage of attendance for every individual.

Input: Put a check on those students in the form (which is automatically generated) who were present or absent, depending on your choice.

Output: Will calculate and push the notification regarding the attendance on individual student's notice board.

9 Viewing Attendance Status:

Description: Faculties (& TA's) can view the attendance data of every student, while student can only view their individual attendance.

Risk: Low

Effort: Few Person Hours

Precondition: Should have enrolled in the course.

Output: Course wise attendance will be shown for every student.

10 Creating Polls/Survey

Description: This will allow faculties to create poll/survey during the semester, so that they can get feedback for their respective courses.

Risk: Medium

Effort: Would require considerable Person Hours

Precondition: User should be a Faculty of the course.

Input: Prepare questions to be added in the poll/survey.

Output: Notification regarding the poll/survey will be pushed to every students wall who taken the course.

11 Answering Polls/Survey’s

Description: Students can give responses to the polls/surveys

Risk: Medium

Effort: Would require considerable Person Hours

Precondition: User should be Instructor of the course.

Postcondition: The results of the polls/surveys will be updated.

Input: Give response to the polls/surveys.

12 Viewing/Downloading Course material:

Description: Students can access the course page and view/download the course material from the courses page.

Risk: Medium

Effort: Would require considerable Person Hours

Precondition: User should be a TA/Faculty/Student of the course.

Performance Requirements:

- The system should provide high performance so that it can be better operational and easy to use. It is a web based project and hence variables like the internet speed, server load, operating system and the browser that a client uses can immensely affect the performance of the system.
- Documents to be uploaded should be in specified format (if asked to) and it will be made sure that they do not exceed the uploading limit that is set.
- It will be kept in mind that the software is designed to be used on a multi-user system, and hence increase in the number of users should not affect the individual response time of each user. This can be tackled with efficient database design.

Safety and Security Requirements:

- A user can only log onto a system if he/she has a valid user-id and password, account of which shall be maintained in a database stored on the server.
- The system would use a secure database and application server, so that an unauthorized person cannot access, or change the present data. The administrator will be given a secure log-in to the database, with all necessary permissions.
- Every user (widely categorized as Administrator, Teacher, and Student) will be given respective permissions so as to avoid any ambiguities and breach of secured information.
- The anonymity of users should be maintained in polling results.

.

Design Constraints:

- The project would require a web server for its implementation, so for implementing and testing purposes we will be locally hosting our website (or may use free online servers).
- The success of delivery of messages will depend on the network, it may get affected due to congestion.

- The time taken in processing of data or upload or download of files will depend on the speed of internet available.
- The number of simultaneous users our website will be able to handle depends on the server’s capacity.

Software quality attributes:

- Sufficient care will be taken to ensure that our software is easily testable and debug friendly.
- The user interface will be designed in such a way that it will be easy to use.
- The software will be compatible with all the frequently used browsers like Chrome, Firefox, Safari, and IE.
- Interoperability is guaranteed as the application will be cross-platform i.e. it will work on different operating systems.

External Interface Requirements

User Interface:

Describes how the users access and interact with the system. All user interfaces in the software will be implemented in a mark-up language.

Hardware Interface:

Describes the hardware components in the system and how they connect to the system. The software is designed to work while connected to the internet, but it works for a standalone device also.

Hardware Requirement:

The hardware interface required for the application is a PC, Laptop, Tablet, or Mobile Phones which have internet connection and a browser.

Software Interface:

Describes how our website will interact with software components like database management system, SMS API.

Database Management System: All the data of our users will be stored in a permanent database maintained by the server. The database will be implemented on an SQL server.

SMS API: Student and their parents will get message alerts if attendance is below specified threshold by the faculty in their respective courses.

Software Requirement:

The Software Interface should comprise of the following components:

1. Server Side:

- XAMPP
- Development Tools: Dreamweaver, Sublime Text Editor.
- Database Server: MySQL.

2. Client Side:

- Running Operating System
- Web Browser

Communication Interface:

- The site communicates with the application server via the Hypertext Transfer Protocol (HTTP) and TCP/IP protocol.
- Data Transfer Rate will depend on internet connection.
- The server will make use of the PHP scripting language to respond to the request from the client.