

# Integrated eLearning Environment - SRS for Phase I

Open Source Innovation Center, KReSIT IIT Bombay  
PI: Prof. Umesh Bellur

February 18, 2005

# 1 Introduction

This project aims at putting together an integrated e-Learning environment for a university student. This will mainly be used by distance education programs offered by institutes, in order to enable students to avail of the academic facilities from any computer connected to the internet. A streaming mode of operation which can be live or offline will be backed by a rich GUI that integrates lectures, quizzing and live interaction with the rest of the classroom making the user feel that they are part of a virtual classroom in the live mode.

Many of the components of such an environment are available today but not in an integrated fashion and this integration will be one of the main contributions of this project. This document specifies the requirements and constraints of such an environment. It is laid out as follows. We will first discuss the different roles that the system will support followed by a discussion on the actual requirements modularized according to functionality. We conclude with non functional requirements and constraints.

## 1.1 Background

This project is the first in a series of projects aimed at advancing the state of the art of the linux desktop being undertaken in the Open Source Innovation Center at KReSIT. We are aiming to create rich user interfaces that can take advantage of the significant processing power possessed by most desktops today. Our e-Learning environment has been divided into two phases

1. Phase I - Offline Streaming Mode: Here we intend to support a mode of operation which will allow students to view lectures on demand from a streaming server. There is no live class room since the lecture has been recorded earlier. However the elements of taking a course will be supported. The focus of the phase is integration of video streaming client along with the other elements of a eLearning environment such as registration for a course, quizzing etc.
2. Phase II - Live Streaming Mode: In this phase we expect to be streaming a lecture live as it is being delivered by the instructor backed by a live interaction capability that allows students to raise and answer questions and recognize and interact with other students in the virtual classroom.

This document details Phase I requirements only.

## 2 User Roles

The system will be having 3 roles - Instructor, Student and Admin. There will be a set of permissions associated with each role.

1. Instructor  
This role enables the user to offer courses, set quizzes, assign grades to students for courses he is offering.
2. Student  
This role enables the user to register for courses, answer quizzes, view lectures/grades.
3. Admin  
This role enables the user to manage users (create, edit and delete) besides performing

other administrative tasks such as monitoring the system operation, editing system configuration etc.

### **3 Description - System Features**

The set of features has been categorized based on the entity that the feature deals with - courses, quizzes etc. Every requirements is a numbered subsection that has as part of the sub section heading, the role needed to use this feature.

#### **3.1 Administrative Features**

The system will come with a default administrator account whose password can be set on installation (similar to the root account on UNIX).

##### **3.1.1 Obtaining User Accounts - All**

Potential users will be able to request for and obtain “user accounts” on the system. A valid email account is a prerequisite to obtain an account. A request for the user account will be registered and be available to the administrator for processing when he logs on. Any user with an administrator role can handle user account requests.

##### **3.1.2 Creating User Accounts - Admin**

Administrators will be allowed to create users and assign one or more roles. A request to create a user needs to be available only if there are requests pending. The creation of a user should close the pending request for creation. The creation will result in the user login and password to be emailed to the user at the mentioned users email account.

##### **3.1.3 Disabling Accounts - Admin**

Administrators can disable users. A account that is disabled will prevent the login from working till it is reenabled by the administrator. Disabling and enabling accounts shall send the user a email to the preferred address informing them of the same. The email should contain a reason for disabling that is supplied by the administrator when disabling the account.

##### **3.1.4 Resetting Passwords and Editing Roles - Admin**

Administrators can edit user details and change the password of the user. They can also remove or add roles to the account. Any change made should result in an email being sent to the user.

##### **3.1.5 Deleting Users - Admin**

Administrators can remove users entirely from the system provided the user has NOT registered for or taken completed any course previously.

##### **3.1.6 Logging In - All**

The system will allow users to login with a password if they already possess a valid account on the system. An invalid password will signal an error. The system will protect password an other user information and ensure that it will not be viewable by others except the administrator.

### **3.1.7 Editing Account Information - All**

Users can edit information themselves and change the password once logged in. Any change saved in the system will be confirmed via an email sent to the users email account.

### **3.1.8 Concurrent Logins**

The system will preclude concurrent logins with the same login name.

## **3.2 Course Creation, Modification and Deletion**

### **3.2.1 Course Creation - Instructor**

Instructors can set up new courses in the system. A course consists of a name given by the instructor, a code assigned by the system that is dependent on the department that the instructor is associated with, a syllabus and a grading policy section that specifies the number of quizzes and assignments that form part of the course evaluation and a set of prerequisite courses (if any). There is also an explicitly identified cutoff point (say a lecture number) beyond which the student cannot drop the course. A course can also be explicitly disabled for registration. Creating the course initially enables or open it for registration. Courses are composed of a set of lectures, quizzes and assignments.

### **3.2.2 Displaying Enabled Courses - Instructor, Student**

The system will display list of courses offered (both enabled for registration and disabled) / registered for if the user has logged in as a instructor or student respectively. The system will display the details of a course to the user, when he clicks on a specific course. Details specified to the instructor include the list of current registrants for the course.

### **3.2.3 Modifying Course Details - Instructor**

If authorized, the user can modify details of the course or delete the course from the system. In case there have been any registrations for the course even if complete, a course can merely be disabled for registration but NOT deleted from the system.

## **3.3 Course Registration and Adjustment**

### **3.3.1 Registering for a Course - Student**

A student is allowed to register for a course that is open for registration. This is done by browsing the list of courses available for registration for which the student satisfies the prerequisites. A course can be selected from this list and registered for. A successful registration results in an email being sent to the student who can then begin viewing lectures.

### **3.3.2 Dropping a Course - Student**

As long as the student has not passed the adjustment point, he can drop the course from the list of registered courses. The system does not need to maintain a history of add/drop activities.

## **3.4 Lectures**

### **3.4.1 Uploading Lectures - Instructor**

The instructor for the course can upload lectures for a given course. At the time of upload, an ordering must be specified by the instructor which says what lectures, assignments and quizzes comes before this lecture can be viewed. This ordering can be modified provided a student having already viewed this lecture will not violate the changed ordering.

### **3.4.2 Viewing Lectures - Student**

Students can view the set of lectures associated with the course including the status of each one (viewed or not). They can then choose to view any lecture that is available for viewing based on their progress in the course. The lecture would then be streamed to the student. For optimization purposes, the lecture media files may be downloadable onto the client's desktop with the appropriate protections that will prevent unauthorized viewing or distribution of the lecture material.

## **3.5 Quizzing**

### **3.5.1 Setting a Quiz - Instructor**

The system will enable a instructor to set a quiz, along with specifying quiz parameters like type of quiz, duration, total marks, prerequisites for taking the quiz and number of attempts allowed. There is also an ordering imposed on quizzes relative to other quizzes, lectures and assignments. A quiz is a set of multiple choice questions. Setting a quiz would imply the instructor also needs to indicate the correct choice for automated evaluation.

### **3.5.2 Modifying and Deleting a Quiz - Instructor**

The system will enable instructor to modify or delete any quiz for a course that he is offering, provided no student has attempted the quiz by then.

### **3.5.3 Taking a Quiz - Student**

The system will enable a student to take a quiz, if he satisfies all the prerequisites for the quiz. The list of quizzes that he is enabled to take for a particular course will appear on the client display from which a quiz can be selected for answering. The system will evaluate responses to the quiz questions online. The system will display results at the end of the quiz, along with correct answers, for all the answers marked incorrect.

### **3.5.4 Retaking a quiz - Student**

The system will allow a student to reappear for a quiz, in case the connection is lost when the quiz is in progress.

## **3.6 Student History and Grades**

The system will store history related to a student like: scores secured by the student in quizzes he has previously taken, the time when he took the quizzes, details of previous attempts of quiz.

### **3.6.1 Viewing History of Courses Taken - Student**

The system will enable a student to view his history and grades for the courses for which he is registered.

### **3.6.2 Assigning and Viewing Grades for a course - Instructor**

The system will enable an instructor to view grades of all students who have registered for any of his courses and also assign grades to the students.

## **3.7 Assignments**

### **3.7.1 Setting Assignment - Instructor**

An instructor can upload assignments to be carried out by the students. Like for quizzes and lectures an ordering of an assignment relative to other assignments, quizzes and lectures will have to be specified.

### **3.7.2 Answering an assignment - Student**

The system will enable a student to upload answers assignments for courses for which he is registered provided the assignment is enabled based on the specified ordering criteria.

### **3.7.3 Evaluating an Assignment - Instructor**

The submission of an assignment will trigger an email to the instructor who can then retrieve the submission and grade it appropriately. Grades will have to be entered manually through a system interface.

## **4 Streaming Requirements**

### **4.1 Synchronization**

Supporting streaming synched with a powerpoint presentation that unfurls as the streaming happens in time.

### **4.2 Resolution**

Having multiple resolutions depending on the speed on the connection from client to server.

## **5 Non Functional Requirements**

### **5.1 Installation**

The client installable will be available on the internet. It will be a complete install (i.e. with no prerequisites). Incremental updates will be available through webstart.

### **5.2 Platform**

The system will be developed on linux platform and will be entirely open source. The system needs to work out of the box on Suse and Redhat desktop and server side versions of Linux.

## **5.3 Performance**

### **5.3.1 Response times**

No operation (defined as a single request/reply) other than streaming of the lecture material will have a response time of more than 5 seconds.

### **5.3.2 Concurrent Users**

The system shall support upto 200 concurrent users with 20 concurrent streaming requests for lecture viewing.

## **5.4 Error Messages**

Every error shall have an appropriate error message. The set of these messages shall be externalized into the file which is read upon startup.

# **6 Constraints**

## **6.1 Client requirements**

The application is expected to have a rich client that will require significant compute resources. The minimum client configuration will be a P4, 2.0 GHz with 512M of RAM and 40G storage.

## **6.2 Connectivity**

The client will have to stay connected with the server via a reasonable bandwidth link for the duration of its interaction. Client will be provided with multiple options to select appropriate resolution according to the capacity of the link with which he is connected to server.