Project Name: WhiteBoard

Team Name: Play-Doh

Kip Nichols ( Liaison)

Jordan Nelson

Kathryn Mitchell

Kathleen Cadenhead

Mentor: Justin Thomas <[justin.thomas@parivedasolutions.com](mailto:justin.thomas@parivedasolutions.com)>

Communication: Asana, GroupMe, Skype

Environment: Linux Mint, Aptana Eclipse Plugin, Git Repo via GitHub

Work Location: Lab

**I. Introduction**

Our project goal is to create a first rate Learning Management System (LMS) for teaching Computer Science courses that is intuitive to use and can be easily extended by future development teams.

**II. Project Overview**

The purpose of the project, which we named WhiteBoard, is to create an easy to use LMS web application built on top of the Ruby on Rails web framework for both professors and students as well as teacher assistants. The application should provide core features such as assignment agnostic uploads including student programs with automated compilation and execution, gradebook functionality, professor and student correspondence, and attendance record keeping. The application should be supported on mobile devices such as the iPad and smart phones in addition to normal web browsers on personal computers.

**III. Business Needs**

Computer Science professors at Baylor University currently do not have an organized and intuitive system for handling assignments, uploaded student programs, grading, and communication with students in a contextual manner.

On the current system, assignments must be uploaded with the exact name given by the professor including the file type, and the number of files specified. A student cannot upload a random number of files unless a compressed folder is used (e.g. zip or rar). Student programs are also difficult to deal with since the professor many times has to download each program and run it manually. Most gradebook functionality on Blackboard, Baylor's default application for grading, quizzes, attendance etc., is not intuitive and has several caveats that can surprise professors that are new to the system.

Communication between professors and students is mainly done through emails, but a lot of time can be wasted when the student fails to include what class they are in, what assignment they are working on, and what issue they are having with the assignment, especially with a software assignment. In addition, taking attendance is painfully slow on Blackboard since professors have to select each student in a drop down box and select their attendance.

We have chosen the following scope shown below because of years of use with the upload site and Blackboard, professor complaints and their suggestions, and our expectations of what we think we can accomplish during the semester.

**Stakeholders**

Our team wants a system that implements many of the scoped features listed below, and demonstrates the knowledge we have acquired over the years.

The Baylor Computer Science professors are really wanting a system that speeds up the process of grading student programs and assignments as well as a way to quickly communicate with students on a specific issue.

Pariveda Solutions mentors and managers are looking to guide our team throughout the project and to see a quality project delivered that meets the needs of its customers.

**IV. Proposal Scope**

**Essential**

The site must have file agnostic uploading with automated program compilation and execution, attendance, gradebook functionality, and public and private communication between students and professors.

**Desirable**

We hope to be able to provide assignment comments, code evaluation, quizzes, discussion thread functionality, a notification system with real time updates, and the ability for users to create groups with shared space for group projects.

**Optional**

Optional features include a calendar with scheduling capabilities (e.g. assignments, meetings etc.), live polling through the website, and a real time chat system.

**V. Limitations**

We have limited time with minimal experience, and a lot has to be learned by each person in the group.

**Questions**

1. In regards to automatic grading of programming assignments (via testsuite), does this include only the output of a program, or the style, spell check of source code, magic numbers etc.?

**Only the output of the program is required, but additional features such as a spell checker on the output and source code would be fine as well.**

2. Is the course correspondence teacher to students, students to students, or both? And is it private, public, or similar to Blackboard messages that teachers send out?

**All are needed: Student to teacher (private), teacher to student (private), and teacher to class (global/public). A Facebook type forum/blog is recommended where the entire class can discuss issues, and comments of comments are recommended as well.**

3. Should we have project assignments that would not be compressed, but would have a directory structure of files?

**Just stick with single files for now, but support multiple file types. Having a directory structure is optional.**  
  
4. Should student registration be done by the students themselves, or should it be done ahead of time by professors or Baylor.

**Students will not register, only professors.**

5. Each class will have either a wall of comments like on Reddit where you can comment on comments, or threads like on a message board. Which is preferred? A good solution may be to simply have topics that each link to a wall of reddit style comments.

**Either is fine, and we plan on doing a forum style of reddit conversations. We will have links with each going to a conversation.**

6. Should we have an email option for any type of notification?

**This would be a nice feature to have, and shouldn't be too difficult to implement.**

7. What do the responsibilities of a T.A. cover? We only know of what they do from the student perspective.

**Solved by the document handed out by Song.**

8. For a professor to add a student, should the professor be expected to just enter the students' BearID/Student ID#? Then our project receive the students' data from Baylor's directory?

**All the components of the project should act like plugins and should be easily extendable, so develop the app so that it expects a well documented format such as: “student\_name, student\_id” so that when the app actually needs to be integrated with the Baylor system, an easy component can be made to take the data from the Baylor servers and convert into this format.**

**Unanswered Questions**

None at the moment.