WhiteBoard

An Efficient and Intuitive Learning Management System

Project Proposal

Noah Christiano Joel Kottas Shir Maimon Jacob Roschen

1 Background

A Learning Management System(LMS) is an interface between students, professors, and school administration. One of the most popular LMSes, especially at colleges and universities, is Blackboard[3]. Blackboard, however, like other LMSes, is extremely slow and has unnecessary features that are often left unused. For example, Blackboard's personalize page features are rarely used, and students avoid the discussion boards. These LMSes designs also often make them difficult to navigate. For example, Blackboard has two sets of navigation tabs, and it is not obvious what the difference between them is. These problems add up to create an interface that is slow and difficult to use, even though there is no need for it to be that way. Students today would rather get what they need quickly than have hundreds of features that they rarely, if ever, use. Other LMSes exist too, such as Moodle[2], Veracross[1], and Canvas[4], but they have their own problems, and none of them are open-source or as lightweight as we want.

2 Objective

WhiteBoard will focus on having an efficient, intuitive interface, and it will have features of excellent quality rather than a large quantity of features. It is open-source, allowing it to be customized and used as desired, although it focuses on the University of Rochester. Most importantly, WhiteBoard will have a unified user experience. It shouldn't take a student who is majoring in computer science three days to find where the grades are posted, and even then to be confused about what their grade is, and it should be easy to navigate to the University of Rochester website. WhiteBoard's design will make such navigation a pleasure.

3 Features

Specifically, WhiteBoard will have the following features:

- Lightning-quick load times
- A secure and painless login system
- Course system, with registration and grades for students
- Schudule and calendar
- Convenient communication between administrators, professors, and students, both for general communication and announcements
- Open source

4 Current Assets

Our team's combined experience covers PHP and Javascript very well, the LAMP stack and all its associated programming languages, and even configuring a Linux server for web hosting. Everyone on the team understands design, and how to learn from both the successes and the mistakes of systems like Blackboard. As students, we have constant access to Blackboard, and we have the expertise to dive into the site to see what code works well and what code doesn't. We are also constantly learning, and our ability in areas relating to this project will only increase.

We are also using several libraries to assist in creating WhiteBoard. We are using jQuery, a javascript library, and code from the open source PHP-Login project, so that we have a secure but simple to use system. There are also many other libraries for Javascript and PHP available should we find them useful.

5 Budget

A large budget shouldn't be necessary. WhiteBoard requires server space on which it can be hosted, but we are planning to get server space on the computer science servers to save money. If we can't get server space there, then we shouldn't need to spend more than $10 per month on hosting.

6 Schedule and Plan

Expected Date Done	Feature(s)
10/23/14	Development environment and datbase structure
10/25/14	Login system and general layout
11/1/14	Courses and grades
11/8/14	Announcements and communication
11/15/14	Schedule/calendar
11/29/14	Presentation

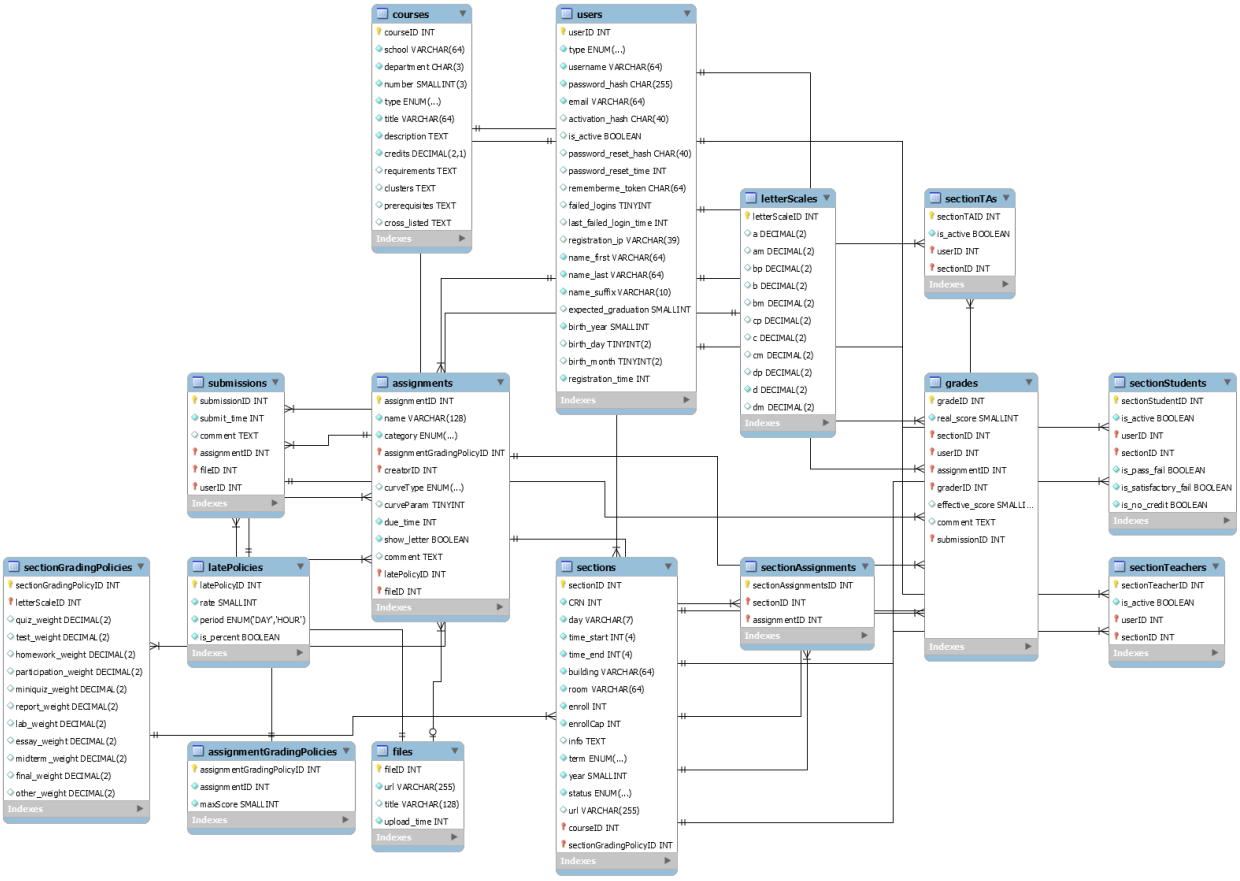
6.1 Development Environment

We are collaborating over GitHub, which combines version control, file sharing, and the ability to edit files online into one service.

We will rent a dedicated server from Online hosting. This Ubuntu server hosts a LAMP stack which is the foundation for the website. The web server hosts two copies of the website, one with a functioning version of the site and another with a test version of the site. This allows us to push changes to the server via GitHub and test our code in real time. The database will use MySQL and the SQL programming language. Our database structure will be similar to the open source moodle project's. However, ours will be greatly simplified in order to promote efficiency.

6.2 Database Structure

We have thoroughly planned the database for the entire system. Fundamentally, the database dictates how the PHP code is organized and implemented. The relationships follow fairly closely with how the back-end will be implemented and integrated. A diagram of the database structure follows.

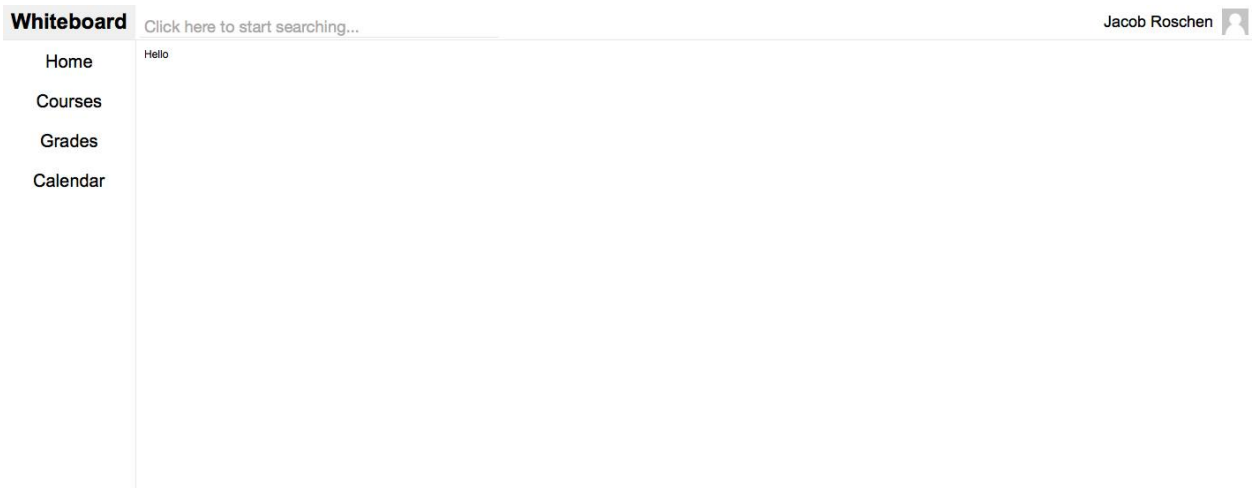


6.3 Login System

With the open-source library we're using, the PHP-Login project, most of the login system is complete. Some adjustments for our front-end and our database structure may be necessary, but they should be fairly small.

6.4 General Layout

The general layout of the page is straightforward: there is a header, sidebar, and the main content area. The goal of this project is not to make anything too fancy, and keeping the user interface simple like this is one of the main ways we accomplish this goal. Below is an image of the current general layout.



6.5 Courses and Grades

Courses will be primarily a way of organizing students with their teachers and their grades. Grades and announcements/communication will go through courses so that they go to the right people. Grades are some of the most important data of the system, but they are fairly simple data. Scaling gets a bit complicated - we plan on implementing it, but this can be removed if necessary for time.

6.6 Announcements and Communication

The primary aim of announcements and communication is for students to be able to contact other students in their classes. For simplicity, this directs to email. It also allows teachers to get a list of their students for whole-class announcements, and for students to get their teachers' emails easily.

6.7 Schedule / Calendar

A schedule and calendar will be part of the site for students to plan their schedules, and to have a general calendar. If time allows, a bonus feature of a schedule optimizer might be added. This section is mostly independent of the other sections, but it involves a lot of front-end. This area has the most flexible features, so we can easily remove functionality here if necessary because of time.

6.8 Presentation

The system should have a clear visual UI and it should be simple enough that it could even be explained with just screenshots and instructions. A the motivation for using our system shouldn't even need a presentation, compared to systems like Blackboard.

References

[1] Breuer & Co. Veracross. https://veracross.com.

[2] Moodle HQ. Moodle. https://moodle.org.

[3] Blackboard Inc. Blackboard. https://blackboard.com.

[4] Instructure. Canvas. http://instructure.com.