

Summary

-Updated: 4-10-2015

-Author(s): Matt, Micheal, Andrew, Carter

ClassBoard is a configurable dashboard for classroom-related widgets. Our goal is to integrate as many classroom interactions as possible into the browser, allowing for a more efficient and streamlined classroom experience.

Users begin by creating either an account with one of two roles-- Student or Instructor. Instructor accounts can set-up classes, enroll students, enable widgets that will be used in the class, input grades, assign homework, and perform all necessary classroom management tasks. Once enrolled, student accounts can personalize their current dashboard and interact with each of the widgets.

Widgets will be the main component of our application. ClassBoard will have widgets for attendance, quizzes, file sharing, questions, homework, grades, lecture slides, and an "I'm confused" button that students can press when they are feeling lost or confused during a lecture. The modularity of the widgets allows for customization and optimization for each individual class and student. Additionally the widgets provide students and professors every piece of information they need during class on their own screen. The widgets also facilitate a more interactive classroom environment.

Revision History

1. First version of document was created

External Libraries:

-Updated: 4-10-2015

-Author(s): Matt, Micheal, Andrew, Carter

- **Express.js:** We're using Express.js because don't want to spend all of our time working on implementing basic http functionality. Express includes a lot of optimization and has extra functionality for HTTP and other things.
- **MongoDB:** Self explanatory; we need a database to be able to store information like registered users, etc. even while the server is not running.
- **connect-mongo:** Middleware for connecting to MongoDB.
- **jQuery:** jQuery facilitates DOM manipulation and event handling and will save us a lot of time.
- **Socket.io:** Allows for fast event-based communication between client and server through websockets rather than ajax. This is vital for our app as we need to update content without reloading the page for fluid dashboard functionality.

- **Bootstrap:** Gives a very good baseline css library and allows for easy manipulation and placement of DOM elements.
- **AngularJs:** Allows for simple and very expandable javascript injection into the DOM element for responsive and dynamically created pages.
- **Passport:** Express middleware for handling authentication and session data.
- **EJS:** Used for creating html templates with javascript logic embedded.

Birds-Eye view

-Updated: 4-10-2015

-Author(s): Matt, Micheal, Andrew, Carter

When a user first loads Classboard they are presented with the login page-- the dashboard and all its functionality require a registered and currently signed in user. The login page has a button which redirects new users to the signup page so they can create a free account. If the user already has an account they can input their information and sign into Classboard. Once the user passes **authentication** they are presented with the specific **dashboard** corresponding to the class they currently have selected. By default this is the class currently in session, or the next class the student has scheduled. Within the navigation bar at the top of the page, the user has the ability to change which class is selected and access a settings menu which will be discussed more later. Inside the dashboard is a collection of **widgets** allowing the user to interact with different aspects of their class. Widgets are dynamic in the sense that when one requires immediate attention, it shifts other widgets out of the way in order to assume a prominent, front stage position on the dashboard. The widgets also change dynamically when the user interacts with them, and some widgets change when the class administrator performs an action. For example, one widget we will implement is an in-class quiz widget (similar to iClickers). The quiz widget will interact with the students by accepting a multiple choice answer and submitting the answer to the server. Once the teacher has closed the quiz, the students' quiz widgets will change visually to indicate the correct answer.

Component-By-Component Breakdown

-Updated: 4-10-2015

-Author(s): Matt, Micheal, Andrew, Austin, Carter

Authentication

-Updated: 4-8-2015

-Author(s): Matt, Micheal, Andrew

-Developer(s): Matt, Austin

The authentication module handles creating accounts, authenticating users, storing user sessions,

and verifying authentication and/or Authorization on each express request. No features of Classboard are usable without first logging in, so the authentication component is closely related to all other components. Authentication is handled by verifying the user's school email and password using the express passport middleware. Each user's password is stored as a unique hashed value that can only be generated using the user's password. This way the user's password does not have to be stored in our database. When a user is authenticated, a session is generated for them and returned to their client. Upon every request the client makes it must include session data which is then used to authenticate the client for that request. Session authentication is also handled using passport. Session data is stored inside our MongoDB instance using the middleware connect-mongo.

Dashboard

-Updated: 4-10-2015

-Author(s): Matt, Micheal, Andrew, Carter

-Developer(s): Carter, Micheal: *UI, drag-and-drop*; Andrew, Matt, Austin: *backend and database functionality*

The dashboard is our application's primary page. The dashboard page is the page that displays each of the widgets for a selected class. The user will have a separate dashboard for each class that they are enrolled in, and if the user is not enrolled in any classes, they will be prompted to enroll before being able to create and customize their dashboard. If the user is already enrolled in a class, they will be brought to the dashboard of whatever class is scheduled next for the student (aka the next class they will attend). Here, users will be able to view and interact with each of the widgets that are available to their class. The user will be able to customize the look of their dashboard by dragging and dropping the widget modules to a desired location on their dashboard screen. We will implement this interactivity with the Sockets.io library. The page will be closely linked with the users database in order to keep track of both the courses the user is enrolled in, and the saved dashboard layout for each class.

Widgets

-Updated: 4-10-2015

-Author(s): Matt, Micheal, Andrew, Carter

-Developer(s): Carter, Micheal: *UI, displaying data in database*; Andrew, Matt, Austin: *storing widget information/preferences*

All widgets share some basic functionality. All widgets have a settings page that can be accessed by pressing a button in the top-right corner of the widget. Once this button has been pressed the body of the widget is replaced with its settings screen. Most widgets interact differently with class administrators than with students, so generally class administrators will be able to change more settings than a student. Widgets all also have separate functions and views for students than for class administrators. For example: the attendance widget will allow students to mark themselves present, but will allow administrators to see the results of the attendance. Widget functionality will be implemented client side using an angularjs directive. Each widget will have its own unique set of api endpoints (/api/widgets/widget_name). Each widget also must have a settings endpoint that will

expose and allow changes to be made to its settings. Finally, widgets will have an angularjs service allowing them to expose public functions to other widgets if needed.

Attendance widget

-Updated: 4-8-2015

-Developer(s): Carter, Michael

The attendance widget will allow students to mark themselves as present and will allow administrators to view the results of the attendance. This will help save time at the beginning of class, and replace the traditional, more time consuming "roll call" strategy of attendance taking.

Quiz widget

-Updated: 4-8-2015

-Developer(s): Matt, Andrew

The quiz widget will allow the professor to quickly ask the class multiple choice questions. The instructor will be able to close the quiz when everyone is finished, and it will be graded automatically in real time. The instructor will then be shown a chart displaying the breakdown of student answers, and will be given the option to share the chart with the class.

I'm Confused Button widget

-Updated: 4-10-2015

-Developer(s): Austin, Matt

The "I'm confused button" widget will simply be a large button available to student users. Students will be able to click the button when they get lost and need the instructor to slow down or repeat what they just covered. This will help to eliminate the pressure that students feel when they have to raise their hand and ask for the instructor to explain a topic again. The "I'm confused button" will be an anonymous way that students can ask for help without feeling judged or pressured.