Project 4

Classboard Functional Specification

Team tbd Michael Lankenau, Andrew Gora, Matt McNally, Austin Sheppard, Carter Tiernan

Written by Carter Tiernan

Last updated 4-10-2015

Summary

- **Classboard** is a configurable dashboard for classroom-related widgets
- **Our goal** is to integrate as many classroom interactions as possible into the browser
    - allow for a more efficient and streamlined classroom experience

- **Users** have one of two roles
    - **Instructor** users can set-up classes, enroll students, enable widgets that will be used in the class, input grades, assign homework…
    - **Student** users can personalize their current dashboard and interact with each of the widgets

- **Widgets** are the main component of our application
    - Classboard will have **widgets providing helpful functionality** such as: *attendance, quizzes, file sharing, questions, homework, grades, lecture slides, and an "I'm confused" button*
    - Widgets are **modular** allowing for **customization** and **optimization** for each individual class and student

External Libraries

- **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.
- **[database]** Self explanatory; we need a database to be able to store information like registered users, etc. even while the server is not running.
- **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.
- **EJS** Used for creating html templates with javascript logic embedded

Birds-Eye View

- Users are **first** shown the **login page**
  - Dashboard functionality requires registered and currently signed in users
  - Users can create an account by following links on the login page
- Once logged in users are shown the **dashboard**
  - The current or next upcoming classes dashboard is shown automatically, but the user can manually select other dashboards
  - The dashboard is a collection of widgets
  - The main navigation bar will allow for users to navigate to a settings page
- **Widgets** are **dynamic**
  - Widgets that require user attention or have been set by the instructor as important arrange themselves on top of less important widgets
  - Widgets have their own settings that can be customized by the user

Component-By-Component Breakdown

Components

1. Authentication and sessions and roles
2. Dashboard
3. Settings
4. **Widgets**
   1. Attendance Widget
   2. Quiz widget
   3. I’m confused button widget

Authentication

- The **authentication** module handles
  - creating accounts
  - authenticating users
  - storing user sessions
  - verifying authentication
  - authorization on each express request
- Since **no features** of Classboard are useable **without** first **logging in** 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
- **Session data** is stored inside our **MongoDB** instance using the middleware connect-mongo

Dashboard

- The **dashboard** is our application's **primary page**
  - Displays each of the widgets for a selected class
- A **separate** dashboard for **each class**
  - If the user is not enrolled in any classes, they will be prompted to enroll before being able to create and customize their dashboard
- Users will be able **to view and interact** with each of the **widgets** that are available to their class
- Users 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.
- 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

- **Widgets** have a settings page that can be accessed by pressing a button in the top-right corner of the widget
  - When pressed body of the widget is replaced with its settings screen
- Most widgets **interact differently** with class **administrators** than with **students**
  - Generally class administrators will be able to change more settings than a student
- Widget functionality will be **implemented client side** using an Angularjs directive

Attendance Widget

- Allows **students** to mark themselves as **present**
- Allows **administrators** to view the results of the **attendance**

- This will **help save time** at the beginning of class, and replace the more traditional, and time consuming, roll call strategy for taking attendance.

Quiz Widget

- Allows **students** to **answer multiple** choice questions from the dashboard
- Allows **administrators** to quickly **ask the class multiple choice questions**, and close the quiz

- Quizzes will be **graded automatically**
- The instructor will then be shown a chart **displaying the breakdown of student answers**
  - The option to share the chart with the class will be given

I’m Confused Button Widget

- Allows **students** to **click the button** when they get **lost**, or **need to instructor to slow down** or **repeat** what he or she just covered.
- Allows **administrators** to see what **percent of the class is still following the lecture**

- Provides an **anonymous** way that **students** can **ask for help**
  - **Removes** feeling pressured or awkward when asking for help