Team Number: May15-31
Project Name: CoderLab
Client / Advisor: Joe Zambreno
Due: 10/13/2014

Week 6 Report (10/7/14 - 10/13/14)

Name (Role)	Team Meeting (10/13)	Advisor Meeting (10/9)	Week Hours	Total Hours
Jake Bertram (Comm)	P	P	4.5	25.5
Dan Smith (Key Concepts)	P	P	4	21
Kyle Tietz (Lead)	P	P	3	21
Jacob Wallraff (Webmaster)	P	P	2.5	20
Erich Kuerschner (Webmaster)	P	P	2.5	21
Bryan Passini (Comm)	P	P	3.5	23.5

Key: P - Present AE - Absent, Excused AU - Absent, Unexcused

Advisor Meeting Notes (10/9)

- Showcased ShareJS demo with Ace Editor component integrated.
    - Considered ways of persisting ShareJS documents
- No word back from CSG regarding the VM
- Shell Streaming: How to handle with multiple users?
    - Zambreno's thoughts: 1 shell per classroom, writable by one person (default: instructor)
    - Limit privileges on the shell user (e.g. create unix user, no root, add some sudoers support if needed)

Team Meeting Minutes (10/13)

Discussion

- ASP.NET vs NodeJS vs other
    - Keep everything the same and go with NodeJS?
    - Possibly too many extra things to set up with ASP.NET (linux + .NET could be a bad time)
- Reorganizing our weekly reports to conform more to rubric
- Assign individual tasks to complete this week
- Half-semester-in-review
    - Good
        - Got a few small demos working to show Zambreno
        - Initial project plan was good
        - Time taken to research projects was helpful
    - Could be better
        - Need to create a more fleshed-out architecture/structure design
        - Need more code output
        - Need better communication about what tasks everyone is on
        - Assign more concrete tasks, organize effort better
- Agenda for next meeting
    - Define project modules and terminology
    - Prepare for design document (10/28)

Individual Accomplishments

Jacob Bertram

- Docker Manager
    - Brainstormed with Dan how to spawn containers and shells
    - From requirements: will have multiple containers, one per “classroom”, with one shell per classroom that one person may use at a time.
    - Drew a diagram, which will probably be a blueprint for the application and will eventually end up in the documentation
- Docker Shells

    - Work In Progress: ability to stream bash sessions over websockets.
    - Currently I've got a bash session streaming to multiple users' web browsers, so its accidentally a collaborative shell anyone can type into. Supports advanced things like vim, password prompts, (n)curses apps
    - Should have a basic demo by next week

Dan Smith

- Worked out the docker manager structure with Jake B.
    - See Jake B's notes
- Learning about JS libraries with Jake B's help
    - ExpressJS web server
    - NodeJS module loading
    - Need to understand how the client-side modules are going to work

Kyle Tietz

- Met with Bryan and Erich to work on ShareJS example
- Considered scalability test with current working example
    - See how performance works across many (20+) sessions
    - Might be able to do it just with multiple tabs in single browser?

Jacob Wallraff

- Going through and learning more about node
    - Using online tutorials/modules as well as materials that Jake mentioned
- Following up with CSG
- Will be absent from Monday group meeting next week (band exhibition)

Erich Kuerschner

- Finished ShareJs demo with Bryan and Kyle
- Added Ace editor to share demo

Bryan Passini

- Finished putting together a ShareJs Demo for our client with Erich and Kyle
- Starting looking into file persistence with ShareJs

    - Save text in editor to a file on the server - one option is to use the file system (fs) module in Node.js
    - Can use doc objects in ShareJs to get the current text from the editor.
    - Opening a file in ShareJs is do able. Get text from the server (which read the file) and use doc.insert() to put into a ShareJs text area. Will probably communicate with the server using browserchannel module in Node.js.

Pending Issues

- Waiting on VMs from CSG
- More research / development is needed to help with planning in the future.
    - Server-side technologies should be considered. NodeJS will be used at the very least for the docker management and ShareJS (if we choose to use ShareJS), but the entire web application's server-side doesn't need to follow this.
- Need a design document with a clear layout of how our technologies will work together and communicate

Plans for Upcoming Week

- Jake Bertram
    - Work on service to spawn multiple docker containers
        - Make webapp that receives code, starts a container, interacts with container, and (eventually) kill the container (no websocket support for now)
    - Help Dan with figuring out the websocket terminal streaming I have started
- Dan Smith
    - Create a demo for the integrated shell and connect with basic code compilation to showcase a program that uses stdin (simple echo program)
        - Use Jake's current mockup as a base

    - Install Jira/Atlassian tools on a server?
- Jacob Wallraff
    - Contact CSG directly about VMs
    - Get up to speed with ShareJS
    - Work with Kyle on ShareJS user sessions
- Kyle Tietz
    - Will set up ShareJS project on Gitlab and add demo we created this week
    - Work with Jacob on ShareJS user sessions
- Erich Kuerschner
    - Improve ShareJS project
        - Add features to the Ace aspect
        - Drop down to select code type
    - Work to send code sendoff to Docker project
- Bryan Passini
    - Add feature to ShareJS project to save ShareJS documents to actual files