Team Number: May15-31
Project Name: CoderLab
Client / Advisor: Joe Zambreno
Due: 10/6/2014

Week 5 Report (9/30/14 - 10/6/14)

Name (Role)	Team Meeting (10/6)	Advisor Meeting (10/2)	Week Hours	Total Hours
Jake Bertram (Comm)	P	P	6	21
Dan Smith (Key Concepts)	P	P	4	17
Kyle Tietz (Lead)	P	P	6	18
Jacob Wallraff (Webmaster)	P	P	5.5	17.5
Erich Kuerschner (Webmaster)	P	P	5	18.5
Bryan Passini (Comm)	P	P	5	20.5

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

Advisor Meeting Notes (10/2)

- Daniel - Worked on checking TogetherJS compatibility with Ace/CodeMirror
    - Todo:
        - Talk with Jake about multiple container management
        - Message Atlassian about a Jira/Stash license
- Erich - Put together ShareJS demo
    - Maybe combine with Jake's docker stuff
- Jacob
    -
- Bryan - Looked at virtual shell options
    - shellinabox - ugly solution for our needs, must be installed on server and requires login

    - jqconsole - shell-style visuals, data handling/streaming would need to be handled by us
- Jake Bertram -
    - Showcased Docker demo - Code can be sent in a POST request and compiled in a listening container, results sent back to user.
    - Todo:
        - Stream docker container's outputs to client
        - Manage multiple containers running at once
        - Challenges: how to manage multiple containers
- Kyle - Looked at available server hosting
    - Decided to use AWS free trial in the meantime

Meeting Minutes (10/6)

Individual Notes

- Jacob Wallraff
    - Talked to CSG, applied for two VMs in case we split up Docker and website
    - Blocked by expired Visual Studio license right now?
        - MonoDevelop?
- Daniel Smith
    - Nada
- Kyle Tietz
    - Got an AWS server up and running
    - Created a NodeJS tutorial project to get familiar with Node and npm
    - Recommend installing ShareJS 0.6.x for best documentation
- Jake Bertram
    - Clarifying requirements - do we need an interactive shell or just code output?
        - An interactive shell or at least the ability to interact with programs reading from stdin is important
    - Looked at managing/handling multiple Docker containers
- Bryan Passini

    - ShareJS exploration, documentation isn't great
- Erich Kuerschner
    - ShareJS stuff

Discussion

What web app server-side technology are we going to use?

ASP.NET - Jacob W.'s most comfortable pick

NodeJS - Jake B.'s most comfortable

tabled this issue for now, team will look into different server-side options

Development Roles/TODO

2 Web app people

- Jacob Wallraff
- Kyle Tietz

2 Docker/shell people - for managing multiple docker containers and streaming shells

- Daniel Smith
- Jake Bertram

2 ShareJS people

- Erich Kuerschner
- Bryan Passini

Group Accomplishments

- Members of the team that are unfamiliar with nodejs looked into it (mostly for ShareJS)
- Two VMs were requested through CSG, received confirmation of request
- Project Plan was turned in, first draft of three.

Individual Accomplishments

Jacob Bertram

- Docker
    - Have a proof-of-concept working! Have a look https://git.ece.iastate.edu/coderlab/docker-server/

        - Basic requirement: *nix system (so Mac should work but I've not yet tested)
    - The README.md files try to explain all the concepts and usage information, let me know what needs clarifying
    - It can build the suitable docker images, launch them with the web service running on port 3000. The server can accept, compile, and run individual C files, fully containerized and isolated from the host system
- Shell
    - Used websocketdjs (npm package) to stream a bash session over websockets
    - Hooked up the websocket with a jqconsole instance on the client-side
    - Things like password prompts, tab completion, vim, anything using (n)curses, etc, will not work and it could be pretty difficult to add support for, because jqconsole is not a full-fledged tty.
        - consider using tty.js (https://github.com/chjj/tty.js/) or some other xterm-compatible emulator
- Offering to help team with NodeJS

Dan Smith

- TogetherJS
    - Not well suited to simple collaborative editing
    - Cursors are not disableable (but can be hidden using javascript/css)
    - User list is not configurable other than name, avatar, color, etc.
    - UI cannot be configured
        - Splash screen always shows upon enabling
    - Works fine with both CodeMirror and Ace, for the most part
        - Even multi-cursor editing works fine
        - (tested in Ace) editing a section that is folded in another client's session will unfold that section
        - Any support for showing other user's cursors/selection would have to be written ourselves using the TogetherJS message system

    - Configuration variables inexplicably have varying effect for different users in the same “room”
- Ace is much easier to configure than CodeMirror
- Todo for next week/meeting:
    - Email Atlassian
    - Talk to Jake about mounting file systems and docker management

Kyle Tietz

- Got AWS running
    - Can make user accounts if others want access
    - Can access linux / window instances via SSH
    - Free payment plan should last for at least a whole month running constantly
- Looked at NodeJS and ShareJS
    - Got tutorial running on AWS
    - Trying to get more complicated examples running

Jacob Wallraff

- Made official website redirect to the one we already created w/ Erich
- Downloaded and installed Node.js, ShareJS
- Experimented with Node.js & shareJS
- Blocked on website progress due to licensing issues with Visual Studio
- Followed up with CSG

Erich Kuerschner

- Finished project web site with Jacob. Now accessible from ece.iastate domain
- Started collaborating with Bryan to get a ShareJS demo together
    - progress looks promising. Goal is to have a demo together by thursday
    - next step is to integrate with Jake docker set-up to send code from sharejs page to a container for execution.

Bryan Passini

- Paired with Erich to build a prototype of text collaboration using ShareJs.

    - Not yet completed - we are still working out a few kinks, but the goal is to have this completed by our meeting on Thursday

Pending Issues

- Do we intend to give users an interactive shell, or simply pipe back the results of running a project? I'm getting mixed messages about it. - Jake
    - Talking with the group clarified things: We want **at least** an interactive console when a user's code is running (e.g., stream the stdin, stdout, stderr of the process when ran) and **at most** to stream a bash session to the user; the difficulty of these is probably identical.
        - How to handle multiple users: Do we spawn individual shells per-user or individual containers per user? (containers per user is impractical for scalability, and a shared shell is impractical with lots of people using it, individual shells might be the way to go)
- 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.

Plans for Upcoming Week

- Continue researching ShareJS and possible web frameworks
- Talk with Zambreno about next steps for project
- Try to square away technologies and begin development