Torey Scheer

Server-side independent study proposal

Spring Semester, 2013-14

# Intro

My project is to make a [trello-like web-service](http://blog.fogcreek.com/the-trello-tech-stack/), while putting my own custom spin on it, such as adding chat functionality on boards, offer additional service-login type (such as twitter), as well as the ability to save/load boards.

# My Goal

My goal is to get comfortable not only using web-services but creating them myself in such a way that would scale in other projects, as well as be easy to implement.

# Why an Independent Study?

While there are other server-side classes, they don’t focus much on web-services, and often require you to use PHP, which I already have a decent understanding from in another server-side class. It will also allow me to work on one project the semester and learn what I need as I work on it, rather than 2-3 smaller projects, and I think this will be good practice for a real job, were I may be required to work on a project in an environment / language I am not totally familiar with.

# What I Will Learn

I would like to create my project in Node.js, as I think this would be a great language to get comfortable with using. I would also like to understand the ins-and-outs of web-services as explained earlier, and I also think it would be good for me to manage myself on a project with multiple different features with a clear-cut deadline.

# Project Goals

* Implement the concepts of “Boards” associated with one / multiple accounts. Accounts can be given different privileges on the boards, such as read, write, “admin”, etc.
* Get “Cards” implemented that are tied to boards
  + Cards can be added to any board, and edited and deleted by all authorized users.
  + Cards can be dragged and placed into sections for organization inside boards. They can also be archived to remove them from being visible on boards.
  + Cards can have messages and images added to them
  + Cards can have a due date, status, and information about who created them and when.
  + Cards can be commented on and rated by users.
* Sockets (for boards and chat)
* Notifications (on-site, possibly email as well) for updates on boards or if someone left you a message.
* Board-specific chats
* Ability to log in using a web-service such as google/twitter/etc.

# Requirements

* Use cookies and sessions for login
* Node.js for server-side
* Database (nosql, mongoDB)
* Sockets (preferably)
* Real-time chat notifications
* User-specific notifications
* Weekly progress meetings
* Final presentation

# Libraries

* <http://coffeescript.org/>
* <http://lesscss.org/>
* <http://jade-lang.com/>
* <http://lisperator.net/uglifyjs/> (possibly)
* Other Node packages as needed (<https://npmjs.org/>)

# Timeline

* **Week 1** – get project set up on server, create a style-less skeleton of the site design for testing.
* **Week 2** – Get database access and login working; create a standard user table.
* **Week 3** – User preferences page for editing password, email, etc.
* **Week 4** – Flesh-out the site’s design and create some functional pages for site testing.
* **Week 5** – Design database for adding “cards” to “boards”, and assigning users to “boards”. Also allow “organizations” that boards can be tied to, that all users in the organization have access to.
* **Week 6 & 7** – Work on “Board” management; editing and deleting “cards”, adding them into “sections”, and allow users to “attach” themselves to a card to get notified.
* **Week 8** – Add comments and priorities to cards, Allow users to search cards they have access to (on any board, or just a specific one).
* **Week 9** – Implement sockets and notifications.
* **Week 10 & 11** – Make card and board management on the client side smoother via JavaScript with dynamic server updates. Server should also send out the necessary notifications to users.
* **Week 12** – Real time chat for boards
* **Week 13** – Extensive testing; try to break and find other potential issues.
* **Week 14** – Polish & bug fixes (code and design).
* **Week 15** – Extras (potentially google login / SSL)
* **Week 16** – Deliverable and final presentation.

# Grading Scheme

* 50% - I’ve completed all the core mechanics (server and client-side)
* 10% - The client looks and feels professionally done.
* 10% - Code is well written (including meaningful variable/method names) and commented (server and client-side)
* 5% - Documentation (at minimum a road-map of the code with brief descriptions)
* 10% - Presentation
* 5% - Weekly progress meetings
* 10% - I’ve gone over and above my original goal / concept

# Needed Resources

* A server that could run Node.js and nosql databases (openshift or heroku).
* <http://stackoverflow.com/questions/2353818/how-do-i-get-started-with-node-js> - would start with many of the resources listed here (I plan on purchasing some books listed here / elsewhere to read over break).

# UI Mockup

