

TEAM HOTSPOTTER

Developers

Dylan Williams
Spence Smith
Nathan Reinhardt

Client

Dr. Igor Crk
Southern Illinois University Edwardsville
Department of Computer Science
Edwardsville, IL 62026-1656

ABOUT OUR PROJECT

Overview

Scan Git based repositories to find potential buggy code based off project contributors and file change frequency

Domain and Business

• Developers

- Tracking bug-fixing commits
- Testing code before pulling into project master

Project Managers

- Assigning troublesome tasks to more experienced developers

Open Source Contributors

- Looking to contribute code to most relevant portions of the codebase

ABOUT OUR PROJECT *(CONT.)*

Core functionality

Scans repositories finding bug hotspots using dynamic metric

- Run from remote server

Provide centralized database to store scan results and repository

- Automatic repository syncing
- Scan result history

Create visualization of repository with heat map

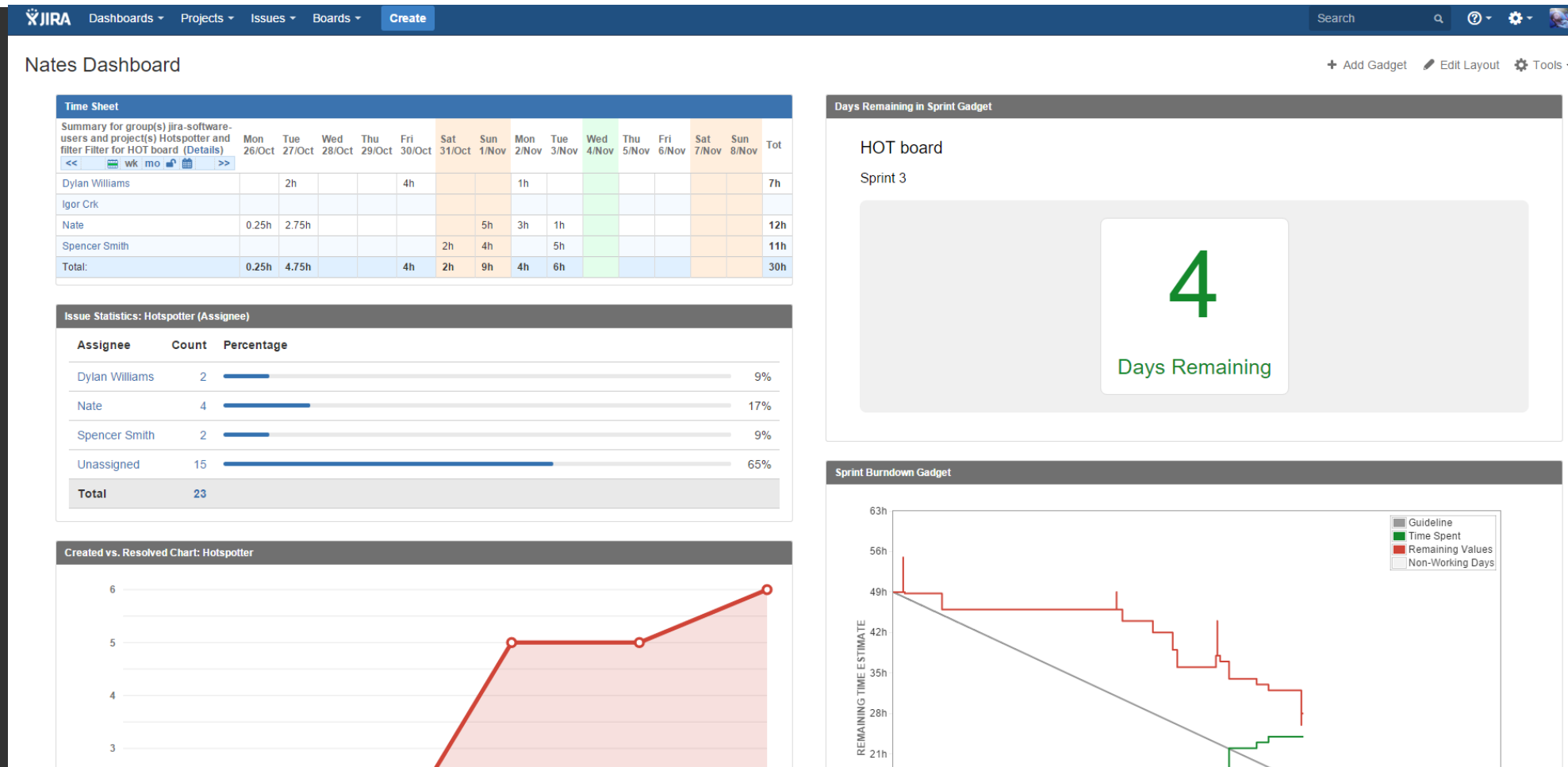
- Filter repository heat map based on author, file, etc.

PROJECT PLAN: MANAGEMENT

- Defined GIT practices
 - Collaboration and responsibility
- Jira
 - Agile Planning
- Jenkins
 - Automated quality/tests
 - Automated deployment



PROJECT PLAN: MANAGEMENT *(CONT.)*



PROJECT PLAN: MANAGEMENT *(CONT.)*

The screenshot displays the JIRA interface for the 'Hotspotter' project, specifically 'Sprint 3'. The board is organized into four columns: 'To Do', 'Elaborate', 'In Progress', and 'Done'. The 'To Do' column contains two main task groups: 'HOT-30' (4 sub-tasks, 'View File Structure') and 'HOT-31' (2 sub-tasks, 'Define Domain Objects'). The 'In Progress' column shows tasks 'HOT-34' (Create Front-End view, 0m), 'HOT-36' (Create API Endpoint, 1h), 'HOT-38' (Domain Docs and Diagrams, 2h), and 'HOT-40' (Create Final Presentation Draft, 2h). The 'Done' column includes 'HOT-35' (Research Angular, 0m), 'HOT-37' (Research Node.js, 0m), 'HOT-28' (Configure Jenkins, 0m), and 'HOT-29' (Status Update #2, 0m). A task 'HOT-33' (Code Review, 5h) is also visible. On the right, a detailed view of issue 'HOT-28' is shown, including its description, 'Given' and 'When' clauses, and a 'Then' clause. The interface also features a top navigation bar with 'Dashboards', 'Projects', 'Issues', and 'Boards' tabs, and a sidebar with various tool icons.

JIRA Dashboards Projects Issues Boards Create

Search ? ⚙️ 👤

HOT board
Sprint 3

4 days remaining Complete Sprint Board

QUICK FILTERS: Only My Issues Recently Updated

To Do

- ▼ HOT-30 TO DO 4 sub-tasks View File Structure
- ▼ HOT-31 TO DO 2 sub-tasks Define Domain Objects
- ▼ Other Issues 4 issues

In Progress

- HOT-34 Create Front-End view 0m
- HOT-36 Create API Endpoint 1h
- HOT-38 Domain Docs and Diagrams 2h
- HOT-40 Create Final Presentation Draft 2h

Done

- HOT-35 Research Angular 0m
- HOT-37 Research Node.js 0m
- HOT-28 Configure Jenkins 0m
- HOT-29 Status Update #2 0m

Hotspotter / HOT-28

Configure Jenkins

Estimate: 5h
Remaining: 0m

Log Work

ISSUE LINKS

0 Add Link

There are no linked issues

Description

0

Given:
I am a developer working on the Hotspotter project
When:
I make a push to master
Then:
I want jenkins to build the newest master branch of Hotspotter and deploy it on the server.

Comments

0 Comment

There are no comments

Attachments

PROJECT PLAN: MANAGEMENT *(CONT.)*

The screenshot displays the JIRA project management interface for the 'HOT board'. The top navigation bar includes 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button. The main content area shows a 'Backlog' board with a search bar and filters for 'Only My Issues' and 'Recently Updated'. The board is organized into a 'Sprint 3' section, which contains a list of issues. The issues are as follows:

Issue ID	Issue Name	Assignee	Estimate
HOT-28	Configure Jenkins	Hotspotter App	6h
HOT-29	Status Update #2	Hotspotter App	3h
HOT-30	View File Structure	Hotspotter App	1d 1h
HOT-31	Define Domain Objects	Hotspotter App	0m
HOT-33	Code Review	Hotspotter App	5h
HOT-40	Create Final Presentation Draft	Hotspotter App	2h

Below the 'Sprint 3' section, there is a 'Backlog' section with 13 issues. The issues are as follows:

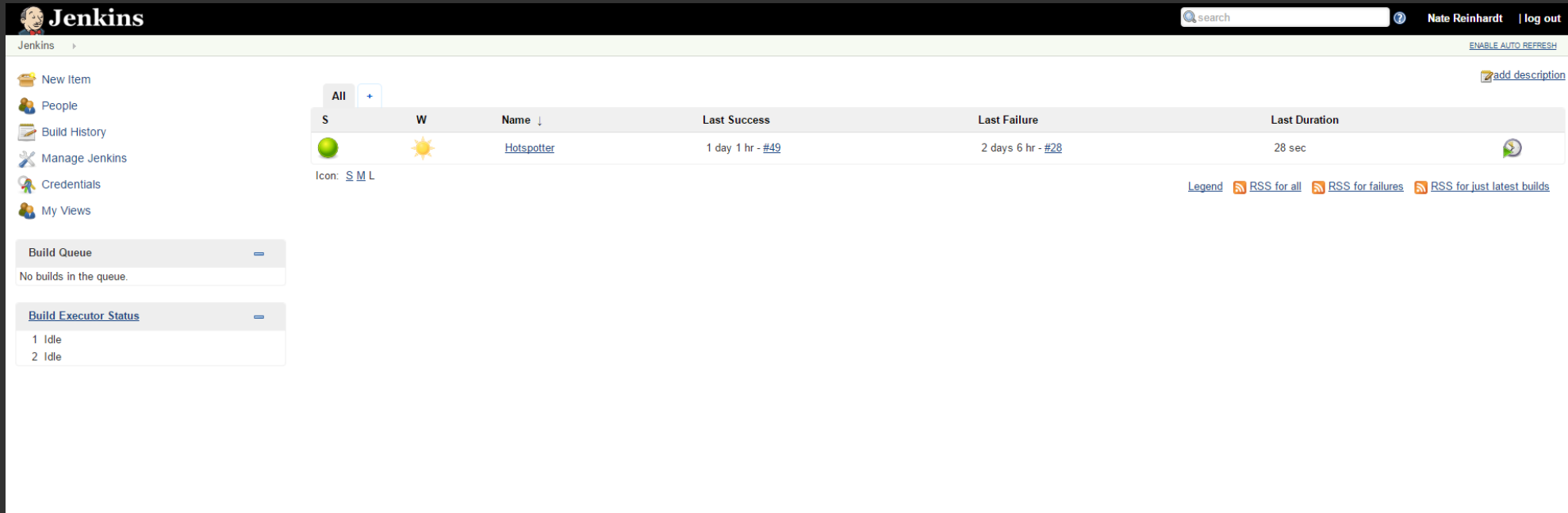
Issue ID	Issue Name	Assignee	Estimate
HOT-32	CSS Optimization	Hotspotter App	6h
HOT-14	Client Log-in	Hotspotter App	20h
HOT-15	Create Account	Hotspotter App	1d 4h
HOT-16	Manage Repository	Hotspotter App	1d 12h
HOT-17	System Sync Repository	Hotspotter App	1d 20h
HOT-18	System Retrieve Repository	Hotspotter App	2d 13h
HOT-19	System Analyze Repository	Hotspotter App	4d 20h
HOT-20	System Store Repository	Hotspotter App	2d 13h
HOT-21	View Hotspot Results	Hotspotter App	1d 20h
HOT-22	Save Result History	Hotspotter App	1d 4h

On the right side of the interface, a sidebar shows details for the selected issue 'HOT-40'. The details include:



- Status:** IN PROGRESS
- Estimate:** 2h
- Remaining:** 2h
- Component/s:** None
- Labels:** None
- Affects Version/s:** None
- Fix Version/s:** None
- Epic:** None
- Reporter:** Nate
- Assignee:** Nate
- Created:** 04/Nov/15 5:59 PM
- Updated:** 04/Nov/15 6:01 PM

At the bottom of the sidebar, there is a section for 'Issue Links' with an 'Add Link' button. The text 'There are no linked issues' is displayed below the button.

PROJECT PLAN: MANAGEMENT *(CONT.)*



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, the user name 'Nate Reinhardt', and a 'log out' link. The left sidebar contains links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'Credentials', and 'My Views'. The main content area displays a table of build history for the 'Hotspotter' job. The table has columns for 'S' (Status), 'W' (When), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. A single build is listed with a green status icon, a sun icon, the name 'Hotspotter', a last success time of '1 day 1 hr - #49', a last failure time of '2 days 6 hr - #28', and a duration of '28 sec'. Below the table, there are links for 'Icon: S M L' and 'Legend'. On the right side of the main content area, there are links for 'add description', 'ENABLE AUTO REFRESH', and three RSS feeds: 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'. On the left side of the main content area, there are two panels: 'Build Queue' showing 'No builds in the queue.' and 'Build Executor Status' showing '1 Idle' and '2 Idle'.

S	W	Name	Last Success	Last Failure	Last Duration
		Hotspotter	1 day 1 hr - #49	2 days 6 hr - #28	28 sec

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle



PROJECT PLAN: RISKS

Tech Stack Knowledge

Mitigated by:

Sharing resources and guide in the project README.
Every time we found something new individually we made an effort to update the readme for other team members to use.

Agile Methodologies

Mitigated by:

- Jira
- Jenkins
- Defined Git plan

I recommend everyone start here
(<https://www.youtube.com/watch?v=AEE7DY2AYvI&list=PLF3T969CR9TJVLgin8tBUlqb27EqQMsi&index=1>)

Back End/NodeJS

(<http://loopback.io/>)
(<https://www.youtube.com/watch?v=GK-tFvpIR7c>)
(<https://www.airpair.com/node.js/posts/top-10-mistakes-node-developers-make>)

Front End/AngularJS

Angular Best Practices/Style Guide
(<https://github.com/johnpapa/angular-styleguide>)

Angular Attribute Binding Explanation
(<https://gist.github.com/CMCDragonkai/6282750>)

MEAN setup guides

(<https://scotch.io/tutorials/setting-up-a-mean-stack-single-page-application>)
(<https://www.youtube.com/watch?v=JnMvok0Yks8>)
(<http://modemweb.com/2014/04/21/mean-stack-a-quick-start-guide/>)
(<http://cwbuecheler.com/web/tutorials/2013/node-express-mongo/>)
(<https://thinkster.io/mean-stack-tutorial>)
(<https://egghead.io/>)

Angular Front End Styling Libraries

(<http://www.google.com/design/spec/style/color.html#>)
(<https://www.google.com/design/spec/material-design/introduction.html>)
(<http://tympanus.net/codrops/>)
(<https://design.google.com/>)
(<http://stackoverflow.com/questions/14741988/twitter-bootstrap-navbar-with-angularjs-collapse-not-functioning>)
(<https://www.codementor.io/angularjs/tutorial/making-a-sliding-menu-with-directives-less-css>)
(<https://material.angularjs.org/latest/>)

Grunt Guides

(http://raibledesigns.com/rd/entry/using_grunt_with_angularjs_for)
(<https://scotch.io/tutorials/a-simple-guide-to-getting-started-with-grunt>)

PROJECT PLAN: QUALITY & TESTS



- Git
 - We defined a Git process to maintain consistency
- Bitbucket
 - Pull requests
 - Full Team Approval
 - Comments on code segments that might have conflicts or opinions

A terminal window with a black background and white text. The window title bar shows 'natem@DESKTOP-O98MG98: /cygdrive/n/dev/hotspotter'. The terminal content shows two lines of commands: 'dev cd hotspotter' and 'hotspotter git:(master) x |'.

```
natem@DESKTOP-O98MG98: /cygdrive/n/dev/hotspotter
→ dev cd hotspotter
→ hotspotter git:(master) x |
```


PROJECT PLAN: QUALITY & TESTS (CONT.)



Pull requests

#43 **OPEN**  → 

Fileview frontend

Overview Commits Activity

Author  **Nathan Reinhardt**


Reviewers  

Description

- initial fileView setup

Hotspotter/client/fileView/fileViewService.js

```
1 +function (angular) {
2 +  'use strict';
3 +  var ngModule = angular.module('hotspotter.fileViewService', []);
4 +  ngModule.service('ParsingService', function () {
5 +    this.ParsingAPI = function(list) {
6 +      var treeData = {folders: [], files: []};
7 +      var tree = treeData;
8 +      // loop through file paths
9 +      angular.forEach(list, function(value) {
10 +        // Split file paths into array and loop through
11 +        var pathSplit = value.replace(/\\/g, '/').split(/,/);
12 +        angular.forEach(pathSplit, function(value) {
13 +
14 +          // ignore '/' root directory
15 +          if (value != '/') {
16 +
17 +            // insert file name
18 +            if (value.indexOf('/') < 0) {
19 +              tree.files.push({name: value});
20 +            }
21 +            // insert directory name
```



111 112 113 B I

This looks kind of funky. Care to explain what is going on?

Comment Cancel

Project Hotspotter



[Workspace](#)



[Last Successful Artifacts](#)



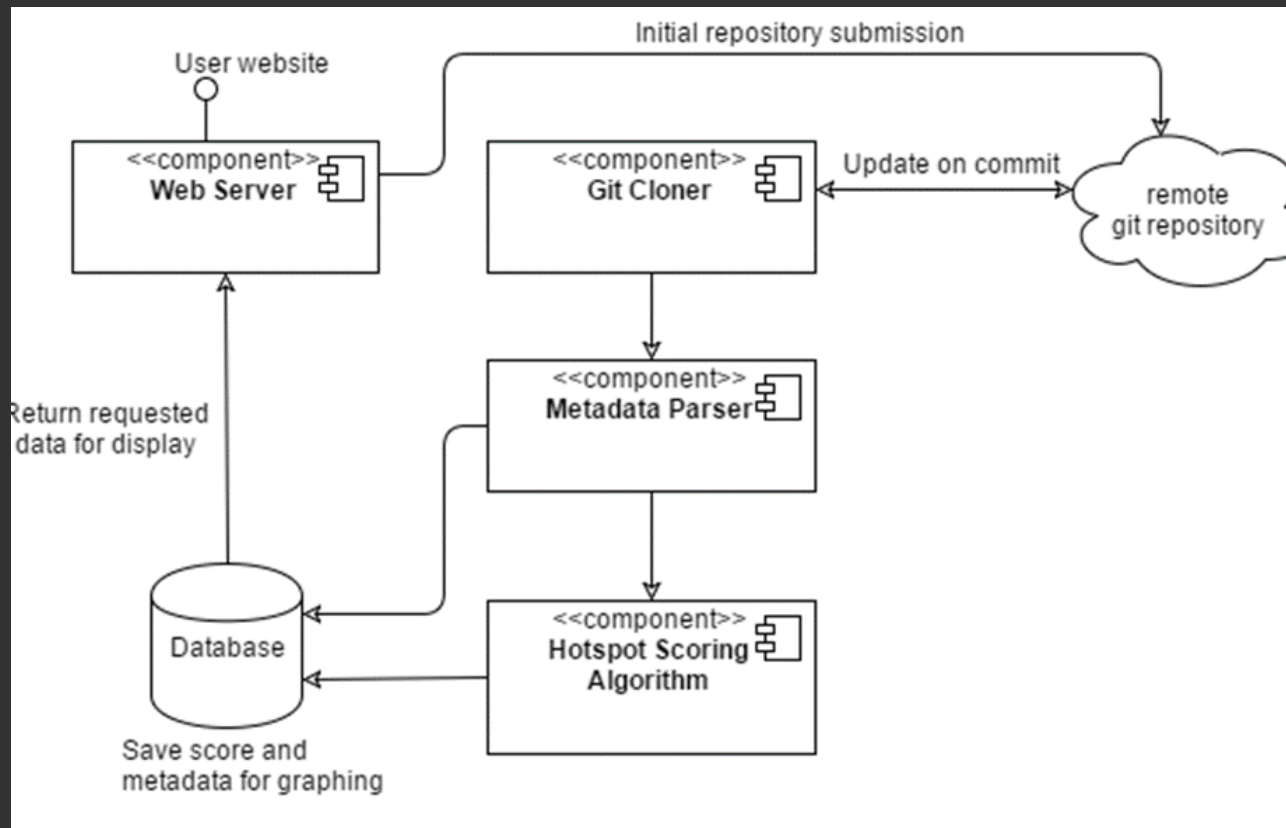
[Recent Changes](#)

Permalinks

- [Last build \(#50\). 3 min 22 sec ago](#)
- [Last stable build \(#50\). 3 min 22 sec ago](#)
- [Last successful build \(#50\). 3 min 22 sec ago](#)
- [Last failed build \(#28\). 2 days 7 hr ago](#)
- [Last unsuccessful build \(#28\). 2 days 7 hr ago](#)

ARCHITECTURE

Client/Server Configuration



MEAN Tech Stack



MongoDB- Database



Express -Middleware



AngularJS – Client Side



NodeJS- Server Side

EFFORTS & DELIVERABLES

- Prototype
- *INSERT GIF/VIDEO OF PROTOYPE*
- *INSERT GIF/VIDEO OF CODE SETUP*

EFFORTS & DELIVERABLES_(CONT.)

- Prototype
- *INSERT GIF/VIDEO OF PROTOYPE*
- *INSERT GIF/VIDEO OF CODE SETUP*

CS425 EXIT STRATEGY

- Met:
 - Working prototype with base/minimal functionality of visualization
 - Familiarization of tech stack of all members of the team
- Was Not Met:
 - N/A

POST-MORTEM

- Good:
 - Processes set-in and adopted by every team member.
 - Great base project to work off of for next semester.
 - Team learned a completely new tech stack.
- Bad:
 - Time planning was slow at first.
 - Didn't get to writing tests for the prototype

CS 499

Next Semester

- Our goal going forward is to use all the processes we set this semester and become a high performing agile team.
- We want to take the existing prototype and start to add scoring algorithms and UI changes
- Implement a dynamic testing suite for existing code.
- Develop with Angular and Node “best practices” in mind.

Questions?