Testing Report

Golf-Matcher

SWENG 894

December 9, 2019

Dan Castellucci

William Ganley

Peter Jester

Craig Roland

Overview

The purpose of this document is to discuss, and report the testing strategies of the Golf Matcher Application

Application Overview

Golf Matcher is an application designed to be your one stop tool for all of your golfing needs. Golf Matcher helps you create teams, compete those teams against one another, and automatically calculate your handicap. Golf Matcher is a completely web based application, and can be accessed anywhere from the office to the greens.

Test Scope

Unit Testing

We performed some level of unit testing for the following modules:

- Util
- Team-add
- Team-edit
- Sidebar
- Authentication
- Teams
- Schedule
- Player-add
- Handicaps
- Players
- Leaderboard
- Players-edit
- Scores-edit

End to End Testing

We performed extensive end to end testing on each module, totalling 57 tests, with 100% passing rate

Performance Testing

Given the scope, and timeline given for the project, we decided that performance testing was not a viable option.

Metrics

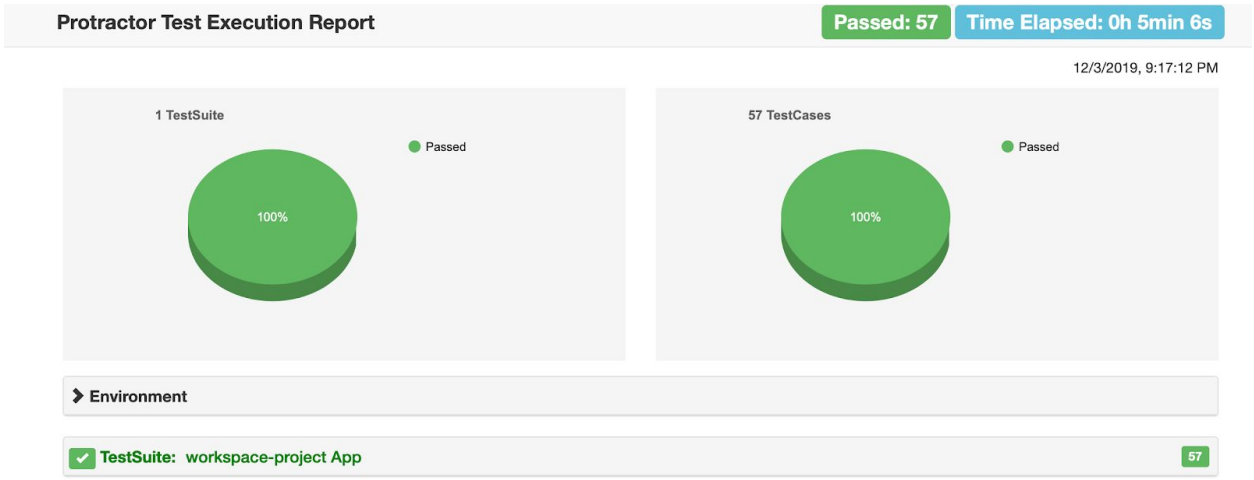
Unit Testing

Statement Coverage: 51.15%
Function Coverage: 44.55%
Line Coverage: 49.43%
Branch Coverage: 3.23%

File	Statements		Branches		Functions		Lines	
src	100%	4/4	100%	0/0	100%	0/0	100%	4/4
src/app/util	100%	16/16	100%	0/0	100%	2/2	100%	15/15
src/app/team-add	84.21%	16/19	100%	0/0	85.71%	6/7	82.35%	14/17
src/app/team-edit	77.78%	21/27	100%	0/0	80%	4/5	76%	19/25
src/app/sidebar	72.73%	8/11	100%	0/0	66.67%	4/6	70%	7/10
src/app	71.88%	46/64	100%	0/0	53.57%	15/28	70.69%	41/58
src/app/authentication	61.11%	11/18	50%	1/2	42.86%	3/7	60%	9/15
src/app/teams	53.33%	24/45	0%	0/6	58.33%	7/12	51.16%	22/43
src/app/schedule	39.13%	9/23	0%	0/4	66.67%	2/3	36.36%	8/22
src/app/player-add	38.46%	5/13	100%	0/0	25%	1/4	33.33%	4/12
src/app/handicap	35.29%	6/17	100%	0/0	16.67%	1/6	35.71%	5/14
src/app/players	32.5%	13/40	0%	0/6	8.33%	1/12	31.58%	12/38
src/app/leaderboard	25.93%	7/27	0%	0/13	12.5%	1/8	25%	6/24
src/app/player-edit	17.86%	5/28	100%	0/0	20%	1/5	15.38%	4/26
src/app/scores-edit	16.13%	5/31	100%	0/0	20%	1/5	13.79%	4/29

End to End

We conducted 57 E2E tests, covering the full span of our application. Below is a screenshot visualizing our E2E testing.

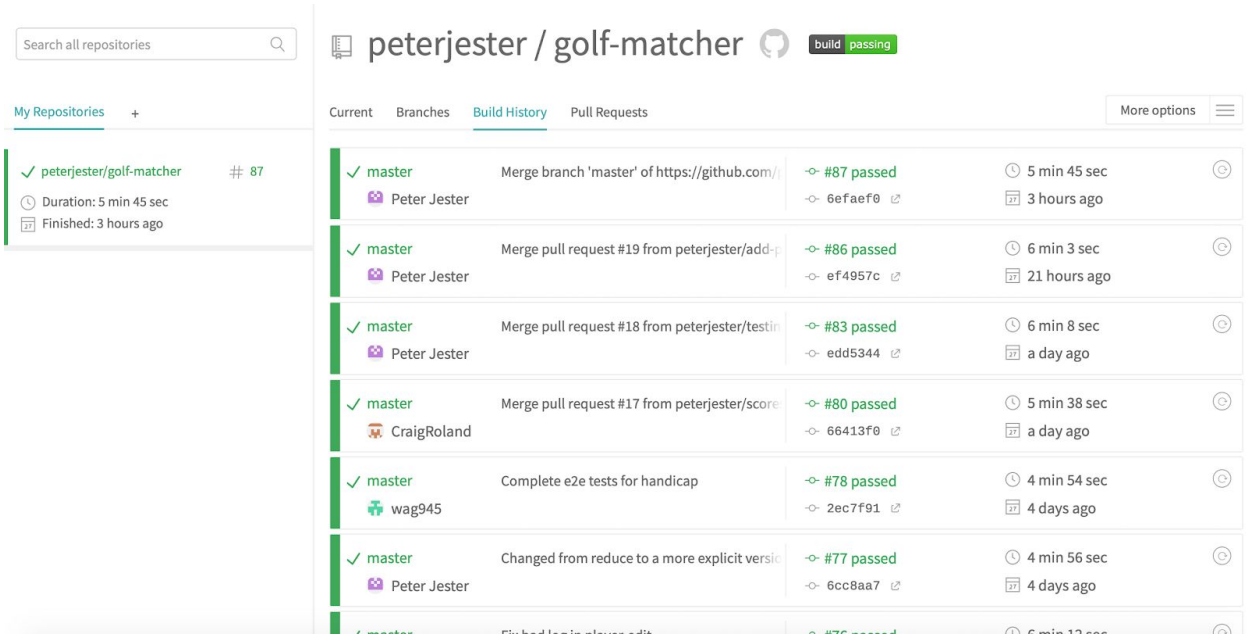


Continuous Integration / Continuous Deployment

In order to ensure that the customers have both the most recent cutting edge features that our team has incorporated, and has the most stable version of the software we are able to deliver, we implement a continuous integration and continuous deployment lifecycle.

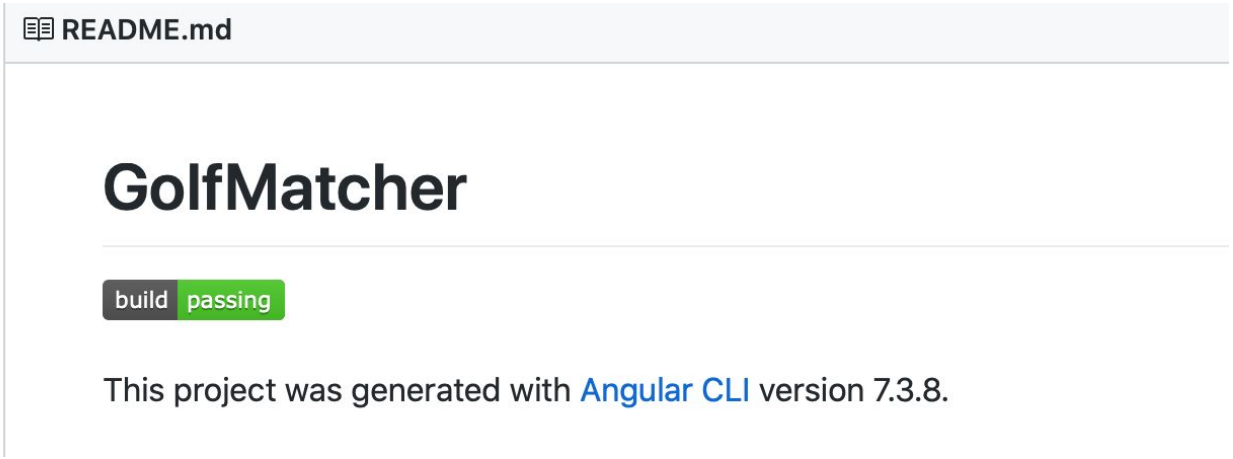
We were able to do this with the help of Travis CI, an open source Continuous Integration tool. We developed a configuration for our application to run unit tests, e2e tests for each individual merge request to master. This ensured that each small to medium code change merged into our baseline, then subsequently deployed, met a certain criteria.

We are also provided with the historical build history of our application as well, so we may identify any patterns. Below is a snippet of our build history.



Additionally, Travis provides us the convenience of automated deployments. For each build that is merged into master, and meets our criteria, it is pushed to our remote hosting solution.

Finally, Travis ensures that we can see the current status of the build at a glance, with a convenient badge that we have adhered to our README, shown below.



Testing Tools

To perform our testing, and continuous integration, we used three tools:

- Protractor - https://www.protractortest.org/#/
- Jasmine - https://jasmine.github.io

- Travis-CI - https://travis-ci.org

Protractor

Protractor is an end-to-end test framework for Angular and AngularJS applications. Protractor runs tests against your application running in a real browser, interacting with it as a user would.

Jasmine

Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks. It does not require a DOM. And it has a clean, obvious syntax so that you can easily write tests.

Travis CI

Travis is a hosted CI/CD tool used for automating testing and deployment for Github software projects.

Lessons Learned

Over the course of this project, we had many take aways when it came to testing an Angular project.

Angular is a new, and dynamically evolving framework. This means that many tools may not be in sync with the latest and greatest Angular version. On more than one occasion, we ran into an issue that did not have an apparent solution, and was not documented. Although we ended up finding work arounds, it was costly.

Conclusion

Testing, of all forms, is vital to a robust software solution. It enabled our team to catch bugs the moment they arose, and kept us all accountable for the quality of our baseline. It was unacceptable to have a broken baseline, and we immediately sprang into action when there was an issue. The issue was always apparent, because we have a timeline of our builds. Overall, testing should be an absolute necessity for any modern software solution.