X Web: Production

Preston Lee

1 Abstract

Your product team has successfully developed what will soon become the first "production" deployment of your database-backed web application. Management has become concerned regarding the over-isolation of individual product teams, however, and as a risk mitigation measure has decided to spread the "tribal knowledge" associated with your product to other teams, and vice versa.

The bulk of the work on this final project will be split into two primary areas:

Deployment You will create a "production" instance of your application and associated MySQL database. In addition, you will work with your external QA team to address any issues they find.

QA You will play the role of QA engineers for a *different* team, and provide them with automated user acceptance and load testing scripts.

You will also need to develop a users guide covering the custom grading criteria you have already defined, and present this work to the class in a "live" manner.

2 Grading breakdown.

All members of your team will receive the same grade for the project, regardless of the fairness of work distribution. Additionally, the team captain will deliver a live demo in front of the class prior to grading.

Total points possible: 36.

20% Final presentation.

20% Automated deployment in collaboration with external QA team.

30% Users' Guide.

30% QA activities for external product team.

2.1 Final presentation.

Your final presentation should contain two parts. First, the entire class–lead by your captain as well as the instructor–will download your Users' Guide and follow the instructions therein. You will have projector access to show slides, code, and other additional resources you deem fit. If you have hard limits on concurrent users, you must specify the constraints in your users guide.

Secondly, you must perform a live demo of the automated user acceptance, profiling, and load testing cases developed by your QA team. Your application will need to respond to the performance specifications you claim in your Users' Guide, though you may do so in a "local", controlled environment.

Quality of content, delivery and project demo will be considered. In other words, make sure your project doesn't "blow up" in front of the class! The entire process should take approximately 15 minutes.

2.2 Automated deployment.

Your team will fully automated the software release process into a two-step process: first making sure the tests pass (must be thorough and cover all custom features, both graded and ungraded), and secondly deploying of new builds as well as running database migration tasks required of the new software. (**Recommended: Heroku's free service tier.** If you have your own server and wish to explore this area in more detail, also check out *capistrano* and *Phusion Passenger*.)

2.3 Users' Guide.

Develop a Users' Guide for your software, walking your users through the core use cases, overlapping with previously-defined custom grading criteria. Please use annotated screenshots where appropriate. You may develop the guide in whatever format you like, provided you also deliver a .PDF rendering with your final deliverable.

2.4 QA activities for external product team.

You will play the role of a QA engineer, and develop the following tests for the external product team to which you have already been assigned:

User Acceptance Tests Use the Selenium framework to automate high-level user testing of all custom grading criteria defined by the external product team, as well as any/all additional use cases covered in their Users' Guide. (Exceptions will only be made for unrealistically implausible test circumstances.) **Important: It is your job to assure your tests are correct and complete, but *not* that they pass! (That is the job of the external developers.)** If the developers have not yet developed a required feature, you may have situations where a test is "completed" before the actual feature. This is not only ok, but expected in some circumstances! These tests should not only "click through" links, but validate that the pages shown are as expected.

Load Testing Using JMeter, create synthetic "users" that put the system under load on your local system. Using the graphs and charts generated from this empiracal data, provide the product team with *real* numbers, as well as the test script(s). If certain components are being problematic from a performance standpoint, let the product developers know! While the use cases should be real, your JMeter file(s) do *not* need to validate system output to the extent that the Selenium tests do.

Bottleneck Profiling For the most resource intensive system operation, provide the external developers with a breakdown of where the bottleneck(s) is/are. It is *not* your duty to fix the issue(s): that work may or may not be addressed by the developers, at their discretion.

3 Project submission.

Use GitHub for all source code, documentation and test cases. Before your final presentation, place a copy of your Users' Guide and any additional presentation materials in the course Dropbox.