CGR Project

Community Games Review

Presentation

Quick intro

Name: Daniel Li

Role: Software Developer

Company: QA Ltd

Product: Full stack game review website

Approach: Implement the MoSCOW method with regards to the MVP condition

MoSCoW

Must have

A front-end web-page which connects to a database via a back-end capable of CRUD functionality. The database must contain at least two entities.

Should have

Users should be able to access links on the web page that allows them to write reviews for specific games, update a specific review and delete a specific review.

Could have

A range of quality of life features such as sort functionality for viewing games on the web page e.g. sort by alphabetical order, sort by aggregate user review score etc.

Won’t have

User authentication.

Tech stack

Version control system: Git

Source code management: GitHub

Scrum board: Jira

Database management system: MySQL

Back-End Programming language: Java

API Development Platform: Spring

Front-End Web Technologies: HTML, CSS, JavaScript

Build tool: Maven

Unit testing: JUnit and Mockito Framework

Integration testing: SpringBootTest

Front-End Testing: Selenium

Testing

Overall testing coverage: % (TBC)

Unit Tests

The unit test files test whether each 'unit' i.e. method of the application is functioning as expected.

Unit tests have been written to cover the following file types:

1. Controller
2. Service
3. Repo

Integration Tests

Integration tests have been written using the SpringBootTest framework, the purpose of which is to test the functioning of the software when modules are combined.

Integration tests have been written to test how the controller interacts with the corresponding service and repo files.

Front end Tests

Tests have been written using the selenium webdriver to test a range of front end (navigation-focused) user stories.

Sprint retrospective

Pros

- Having clearly defined and distinct user stories was really useful for Selenium testing.
- Using smart commits helped keep my git commits atomic and specific.

Cons

- Sprints were siloed between the front and back-end. In future, I feel it would be better to do them concurrently, as when I was writing the front-end, I would then realise perhaps the endpoint needed some modification or it would be useful to have an additional custom query etc.

Live demo

Conclusion

Summary

Created a functioning full stack application which satisfies the MVP conditions.

Implemented stretch functionality associated with queries relating to the two entities e.g. calculating the average score of user reviews for a specific game.

Areas of improvement

Leave more time for testing/employ more test driven development practices - Leaving testing so late in the day, meant I didn't have much time for peer review/debugging.

Q&A