

The background is a solid dark blue. Overlaid on this are several thin, gold-colored lines that form abstract, angular shapes. These shapes resemble stylized mountain peaks or architectural structures, with some lines extending from the edges towards the center. A prominent rectangular frame, also in gold, encloses the main title text.

# Hobby Spring Project

By Andrew McCall

# Version Control & Source Code Management

## AM: Working Dev into main #5



Andrew-McCall wants to merge 29 commits into `main` from `dev`



Conversation 0



Commits 29



Checks 0



Files changed 36



Andrew-McCall commented now

Owner



No description provided.

Andrew-McCall added 29 commits 10 days ago

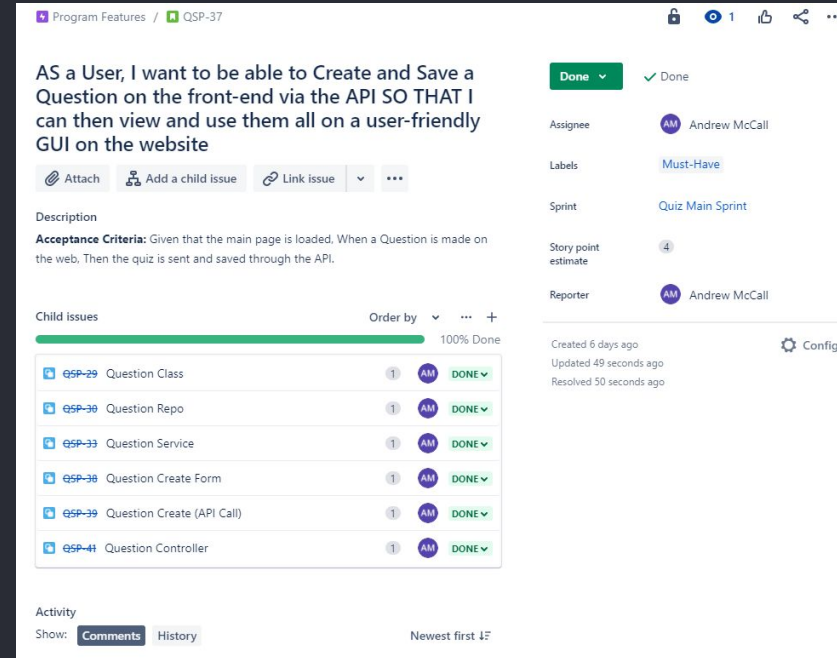
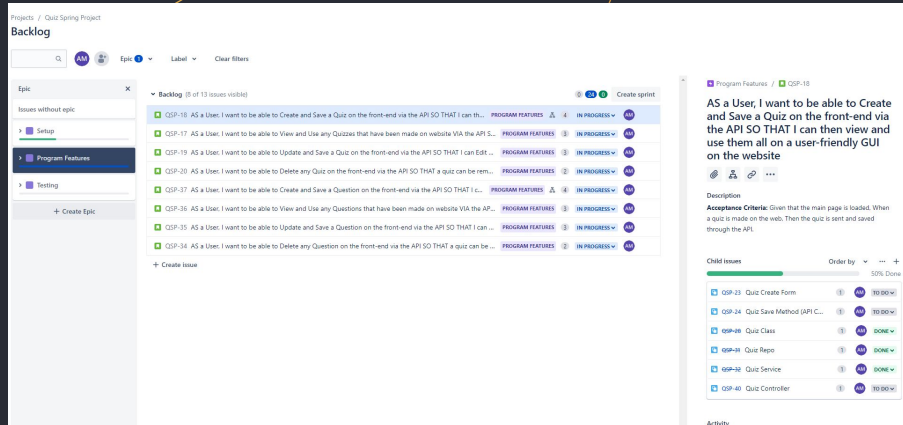
- AM: QSP-28 QSP-29 > Created Quiz & Question Classes 7db22e1
- AM: QSP-30 QSP-31 > Created Quiz & Question Repo interfaces (No Custo... 8ab828e
- AM: QSP-32 QSP-33 > Created Quiz & Question Services 4962173
- AM: QSP-6 > Configured Connection to GCP in app.properties Also fixx... 5343b39
- AM: Controllers Started 200a9ce
- AM: Quiz Controller Started, Create and Get e088a15
- AM: QSP-40 > Quiz Controller 27f014a
- AM: general fixes and clean-up 04a6b40
- AM: Fixed the infinite recursion of the get json API callgit bed4cf2

Github was used to manage and store previous versions of my code.

Github

# Kanban Board

I used jira to layout my project on a Kanban Board with the SCRUM agile Methodology.



# Database

Using free trial credits, I reserve a single core on Google Cloud Platform for my MySQL database to run headlessly in the cloud.

<input type="checkbox"/>	Instance ID <span>?</span> <span>↑</span>	Type	Public IP address
<input type="checkbox"/>	<span>✓</span> qsproject-1	MySQL 8.0	34.105.178.249

Configuration

vCPUs


1


Memory

3.75 GB

SSD storage

10 GB


 Free trial credit



£163

Free trial credit

Out of £216



43

Days remaining

Ends 3 August 2021

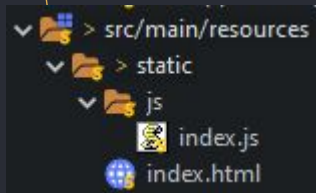
You will not be billed during your free trial. To keep your projects running after the free trial has finished, upgrade to a paid account.

UPGRADE

LEARN MORE

# Front-End Web Technologies

Html files in the static will be caught by Spring Boot and then hosted. There is only one page, index. It uses Bootstrap CSS and Bootstrap Js as well as index.js which interacts with the REST API



Quiz Name

Description

Create New Quiz

Question Set 1

Question description 1

Question	Answer	Edit
Question 1	Answer 2	<div>Edit</div> <div>X</div>

Use Quiz

Edit Quiz

New Question

# Build Tool

```
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ SpringQuizProject ---
[INFO] Building jar: C:\Users\andre\Documents\eclipse\QuizSpringProject\target\SpringQuizProject-0.0.1.jar
[INFO] --- jacoco-maven-plugin:0.8.6:report (jacoco-site) @ SpringQuizProject ---
[INFO] Loading execution data file C:\Users\andre\Documents\eclipse\QuizSpringProject\target\jacoco.exec
[INFO] Analyzed bundle 'SpringQuizProject' with 7 classes
[INFO] --- spring-boot-maven-plugin:2.5.0:repackage (repackage) @ SpringQuizProject ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 24.190 s
[INFO] Finished at: 2021-06-21T11:36:35+01:00
[INFO] -----

andre@LAZIRUS MINGW64 ~/Documents/eclipse/QuizSpringProject (main)
$ java -jar SpringQuizProject-0.0.1.jar
```

Using maven i can package my project into a .jar or .war (depending on the where it will be distributed to (Web Server or General Application)).

Maven runs all my test before it completes any build and if a test fails, the build is canceled

# Automated Testing - Junit5, Mockito, Selenium

My tests are ran automatically on each maven build and on average, they cover 92% of my total code.

The classes with below 80% coverage are made of auto-generated functions that do not need testing.

QuizSpringProject	<div><div></div></div> 92.3 %	1,981
src/main/java	<div><div></div></div> 81.5 %	616
com.qa.quiz.domain	<div><div></div></div> 76.6 %	403
> Question.java	<div><div></div></div> 72.0 %	198
> Quiz.java	<div><div></div></div> 81.7 %	205
com.qa.quiz.rest	<div><div></div></div> 87.2 %	82
> QuestionController.java	<div><div></div></div> 87.2 %	41
> QuizController.java	<div><div></div></div> 87.2 %	41
com.qa.quiz	<div><div></div></div> 37.5 %	3
> SQPApplication.java	<div><div></div></div> 37.5 %	3
com.qa.quiz.service	<div><div></div></div> 100.0 %	128
> QuestionService.java	<div><div></div></div> 100.0 %	64
> QuizService.java	<div><div></div></div> 100.0 %	64
src/test/java	<div><div></div></div> 98.2 %	1,365

```
@Test
void CreateQuestionTest() throws Exception {
    Question question = new Question(new Quiz(), "Changed Q", "Changed A");
    Question created = new Question(2L, new Quiz(), "Changed Q", "Changed A");

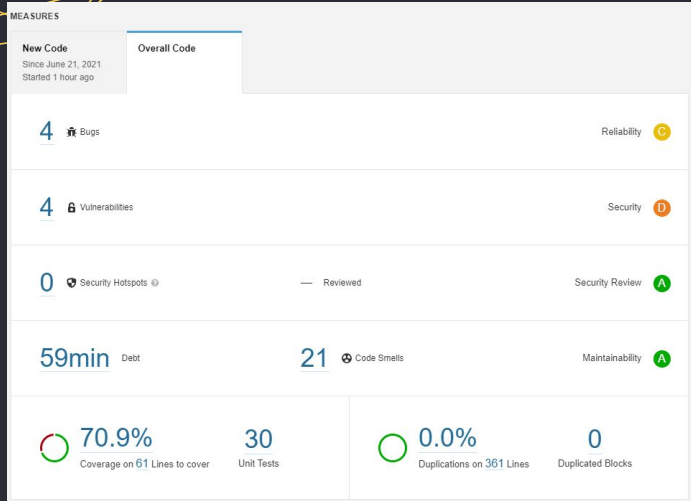
    String JSONQuestion = this.mapper.writeValueAsString(question);
    String JSONCreatedQuestion = this.mapper.writeValueAsString(created);

    RequestBuilder mockRequest = post("/question/create").contentType(MediaType.APPLICATION_JSON).content(JSONQuestion);
    ResultMatcher checkStatus = status().isCreated();
    ResultMatcher checkBody = content().string(JSONCreatedQuestion);

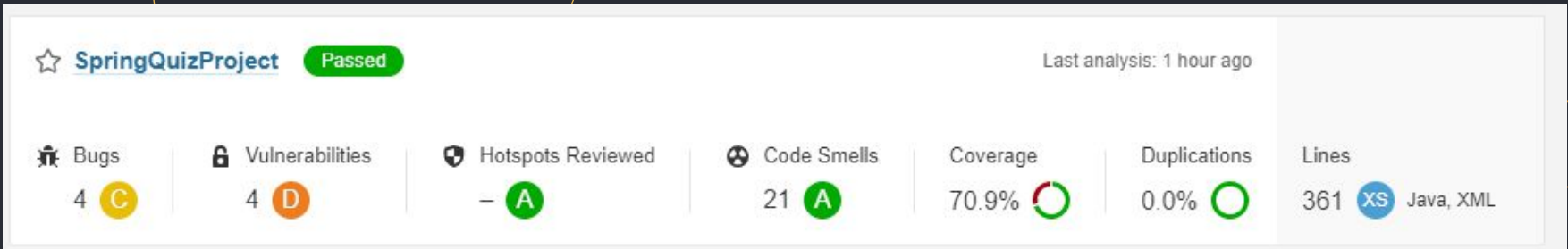
    this.mvc.perform(mockRequest).andExpect(checkStatus).andExpect(checkBody);
}
```

5	8
0	128
0	64
0	64
25	1,390

# SonarQube



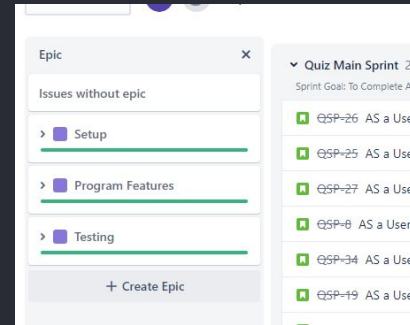
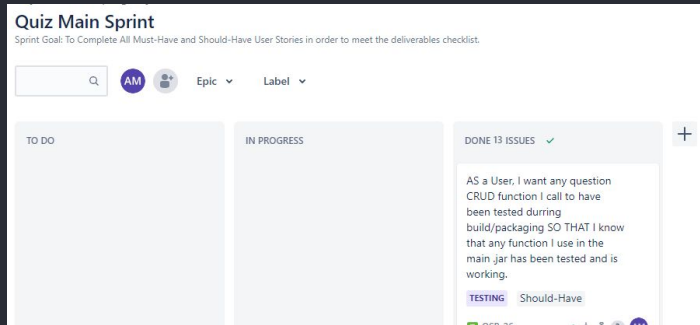
Sonarqube is a tool used to perform automatic reviews/analysis of code to detect bugs, code smells, and security vulnerabilities.





# Sprint Retrospective & Review

Review: I think my sprint went well, All the user stories were completed as well as the sprint goal (To complete all must-have and should-have stories). This meant I finished my project and used every technology expected in the requirements given.



Retrospective: Because my board only contained stories that were must-have or should-have, it meant my project was more limited. Should i of finished early i would have had to restart the planning process and create more user-stories on the spot rather than having all the planning at the start. An example of an improvement would be a more interactable use function for each question set, this would of been good as a could-have as should i of had time, it wouldn't disrupt my workflow to then complete and test for the already created user story rather than breaking the agile cycle if i wanted a new feature.

# Conclusion

Although I finished my project, using the new technologies I'd learn the week prior, I could of started my project with more ambitious user-stories so additional features of improvements could have been implemented more easily without disrupting my work-flow. All my user-stories were must-have or should-haves and therefore equally important.

Git and Jira were the easier technologies I used, not only being relatively simple in use (As they were designed by companies to be user-friendly) but I had also used them in my previous projects.

As sonarQube was mentioned in the spec, i used it for this project for the first time and found it's analysis and features very useful even if it required another service to be hosted on pc while i program and its testing coverage was calculated differently to the coverage from eclipse.