



Food Review Project

QA Project

Salah Salah

Intro

- Who are you?

My name is Salah, I am a QA software consultant and I am currently in my second month of QA training where I am doing my second project which is a food review project of restaurants I have reviewed. I enjoy working as a software consultant, as I have a chance of working with clients and also creating the work they need.

- Approach?

My approach was to create a kanban board to first set out my user stories so I can have a rough idea on what work I will be creating and the route to follow. I then worked on making my github feature branches. From there, I went on to create my codebase and my testing for both backend and frontend code.

Concept

- How was problem approached initially?

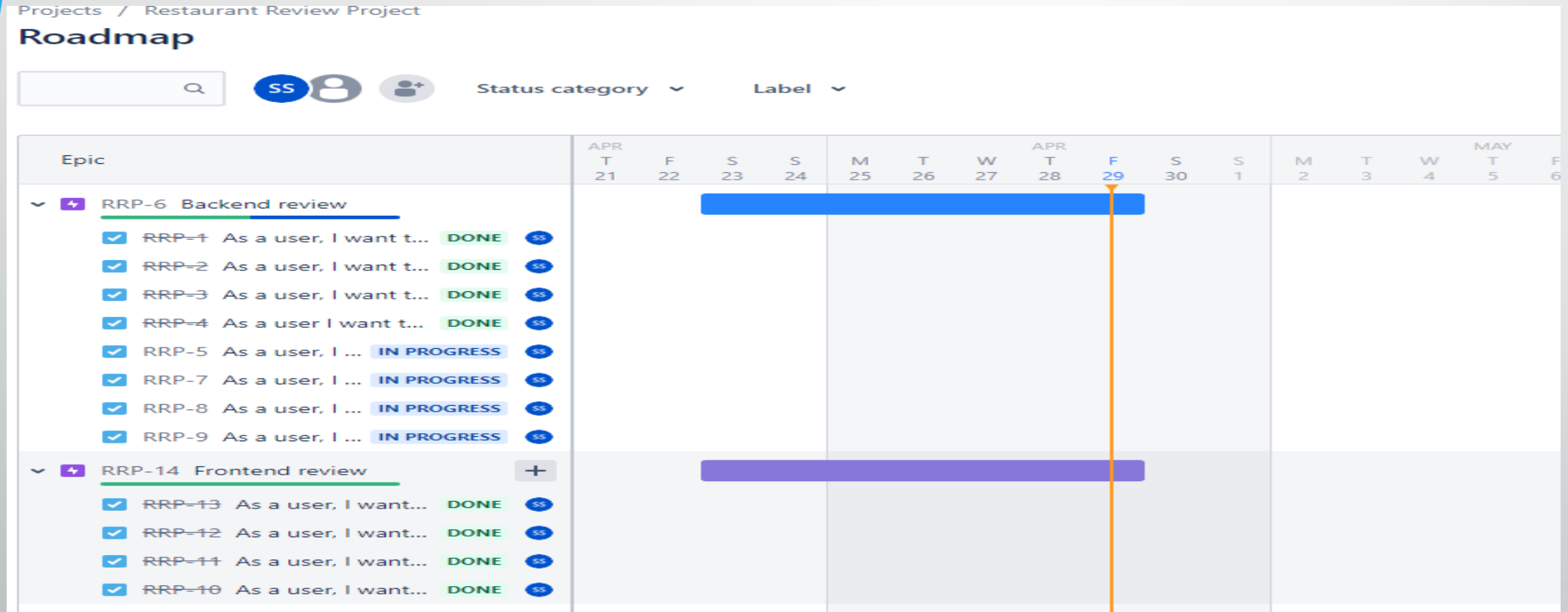
I began by creating a roadmap on jira to show my journey and if there may be any changes needed to do, I can quickly go into my project management board and change it there, before creating code or touching github like previously.

Sprint Plan

- What needed to be included? What did you hope to achieve?

I needed to make sure a solid project management board was implemented, as was a codebase which linked from frontend to backend was working and also have no errors in my code. I hope to achieve a fully integrated API which can be used by others.

Project Management Board



<https://salah456.atlassian.net/jira/software/projects/RRP/boards/3/roadmap?timeline=WEEKS>

Consultant Journey

- What technologies have you learned for the project?

In this specific project I used, Git, GitHub, Jira, MySQL, Java, Spring, HTML, CSS, JavaScript, Maven and JUnit & Mockito



Continuous Integration (CI)

- How did you approach version control?

Learning from my last project, I made sure I created off my develop branch and made feature branches from there. I began making feature branches for nearly all my file classes, such as dto, entity, controller and then branch back to my main branch. I also made two repositories for my front end and back end files so I didn't have any code mixing errors as code errors were more frequent in my previous project

```
guled guled@DESKTOP-AP6VVOG MINGW32 ~/eclipse-workspace/Food_Review_Project (develop)
$ git checkout -b 'controller_feature_branch'
Switched to a new branch 'controller_feature_branch'

guled guled@DESKTOP-AP6VVOG MINGW32 ~/eclipse-workspace/Food_Review_Project (controller_feature_branch)
$ git status
On branch controller_feature_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   pom.xml
        modified:   src/main/resources/application.properties

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        src/main/java/com/qa/food_review_project/controller/
        src/main/resources/application-dev.properties
        src/main/resources/application-prod.properties
        src/main/resources/application-test.properties
        src/main/resources/movie-data.sql

no changes added to commit (use "git add" and/or "git commit -a")

guled guled@DESKTOP-AP6VVOG MINGW32 ~/eclipse-workspace/Food_Review_Project (controller_feature_branch)
$ git add src/main/java/com/qa/food_review_project/controller/

guled guled@DESKTOP-AP6VVOG MINGW32 ~/eclipse-workspace/Food_Review_Project (controller_feature_branch)
$ git status
On branch controller_feature_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/main/java/com/qa/food_review_project/controller/RestaurantController.java
        new file:   src/main/java/com/qa/food_review_project/controller/ReviewController.java

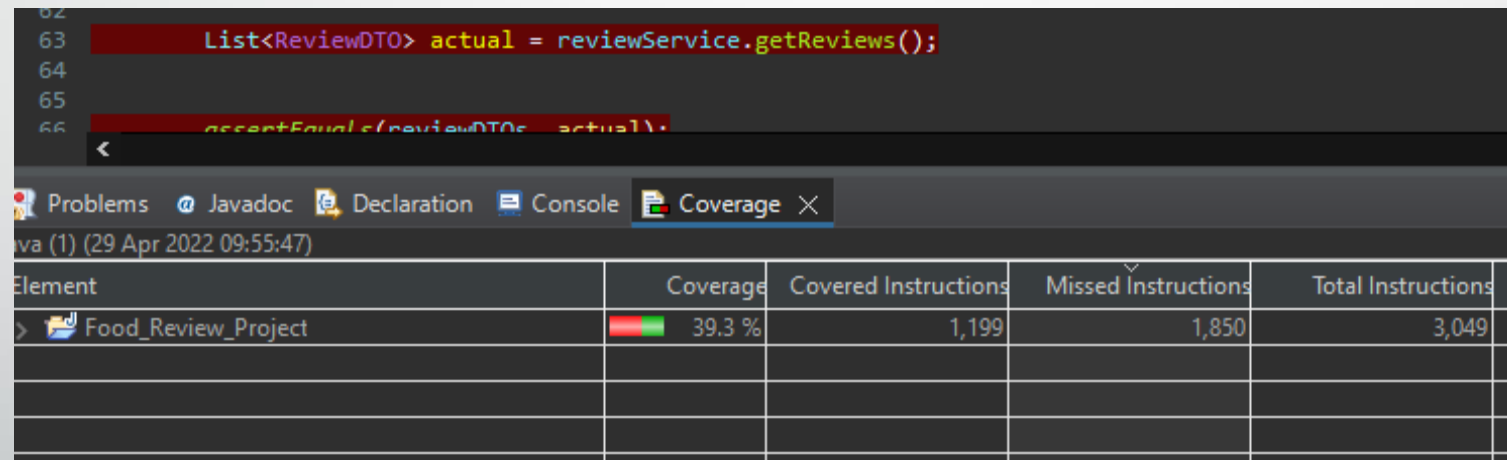
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   pom.xml
        modified:   src/main/resources/application.properties

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        src/main/resources/application-dev.properties
        src/main/resources/application-prod.properties
        src/main/resources/application-test.properties
        src/main/resources/movie-data.sql

guled guled@DESKTOP-AP6VVOG MINGW32 ~/eclipse-workspace/Food_Review_Project (controller_feature_branch)
$ git commit -m "created controller classes"
[controller_feature_branch cf79627] created controller classes
2 files changed, 146 insertions(+)
create mode 100644 src/main/java/com/qa/food_review_project/controller/RestaurantController.java
create mode 100644 src/main/java/com/qa/food_review_project/controller/ReviewController.java
```

Testing

- I am currently still working on errors in my codebase which mainly is coming from my test files for controller. Most of my test files began with lower coverage when I first started testing.



The screenshot shows an IDE with a code editor at the top and a 'Coverage' tab at the bottom. The code editor displays two lines of Java code: `List<ReviewDTO> actual = reviewService.getReviews();` on line 63 and `assertEquals(reviewDTOs, actual);` on line 66. The 'Coverage' tab shows a table with the following data:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
> Food_Review_Project	39.3 %	1,199	1,850	3,049



Demonstration

- Live Demonstration

Sprint Review

- What did you complete?

I completed my backend and spring api codebase, aswell as my project management board which I felt lead me to a great start for this project.

- What got left behind?

Unfortunately, I did not fully complete my front end work but this is as I spent more time working on my backend codebase. I also didn't end up reaching the industry standard of 80% coverage for my testing but however I am still working on it.

Sprint Retrospective

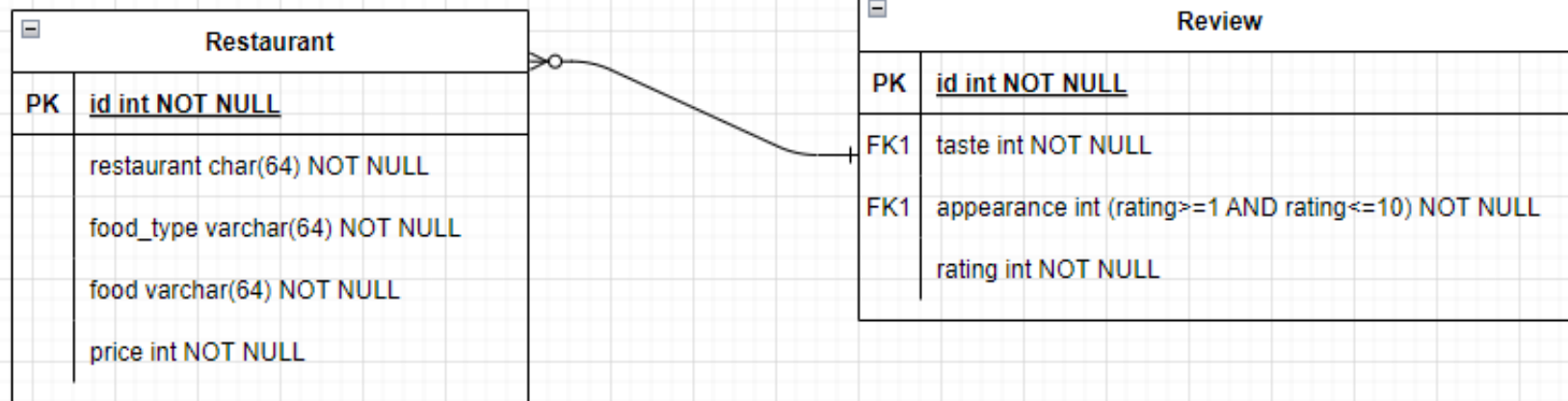
- What went well?

I enjoyed working on keeping my branches in a neat condition as it allowed me to know what the next piece of work was meant to be completed, so it gave me a good roadmap to work off.

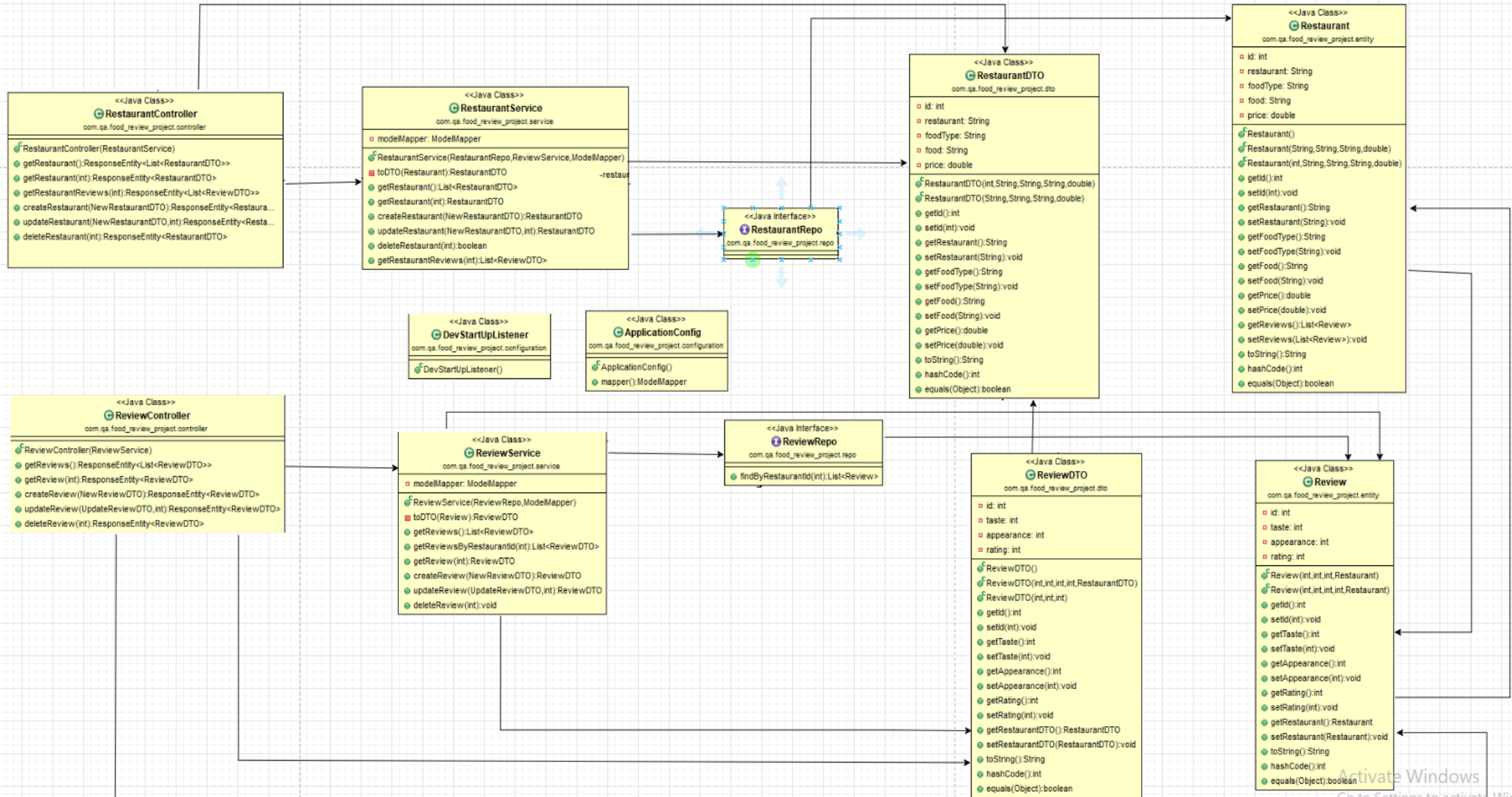
- What could be improved?

I would say it could be splitting the time I spent working on my backend and frontend code more equally so I can have more balanced work instead of rushing more of my front end work.

ERD



UML





Conclusion

- Reflections on project, future steps, any other relevant info?

What I took from working on this project is making sure I take more time understanding the integration between two codebases as it will be vital for future projects I could work on. Also working code errors more time efficiently and no spending endless time trying to fix.



Thank You

- Q&A