

TO-DO LIST PROJECT PRESENTATION

Harry Fresco

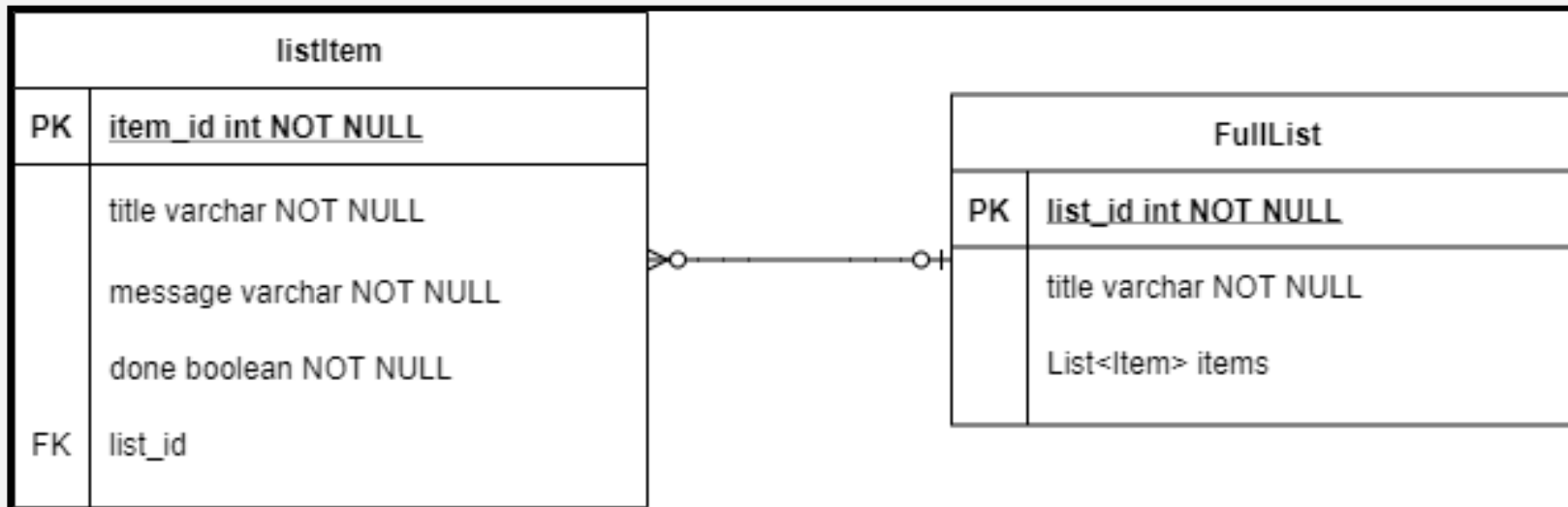
CONTENTS

- Introduction
- Consultant Journey
- CI
- Testing
- Demo
- Sprints
- Conclusion

INTRODUCTION

- Overview of how I approached the specification:
 - User stories and created [Jira board](#)
 - Designed database with diagram
 - Created object classes and their DTOs and Controllers in Spring
 - Created back-end and front-end simultaneously
 - Left more time for front-end development
 - Implemented variety of testing throughout, such as Junit, Mockito, Selenium

DATABASE DESIGN



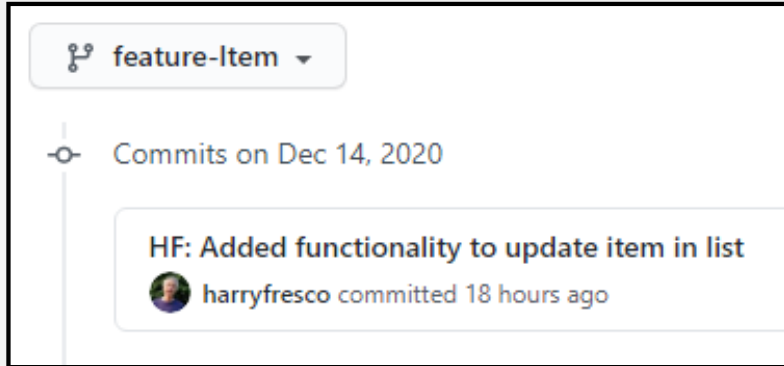
This was the original Database design that remained the same throughout development.

The List Entity had to be called FullList as List was a keyword in Java.

The requirements surprisingly didn't change throughout development.

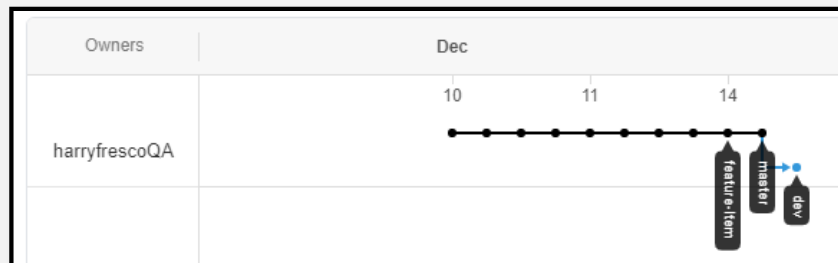
CONSULTANT JOURNEY

- Knew Java, Junit, Maven, Jira, from previous project
- HTML and JavaScript from Uni, but never went well
- Spring, steep learning curve
- SonarQube
- Selenium



CI

- Used Main/Dev/Feature branches
- Main
- Dev
- Feature
 - Used for when a specific feature is being developed and code may not compile
 - Such as feature-item
 - Merged to dev when complete



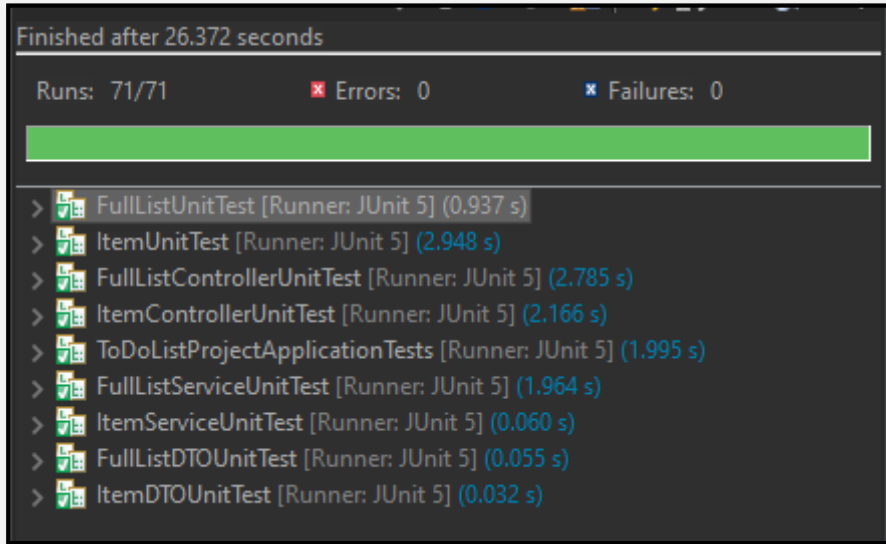
SMART COMMITS

- Occasionally used smart commits to connect to Jira
- Not fully utilised
- Could have used it to move cards from to-do to done etc.

HFTDLP-8 #comment HF:Added back-end support for lists and started web... ...



harryfresco committed 5 days ago



Element	Coverage
▼ To-Do List Project	81.0 %
> src/main/java	81.0 %
> src/test/java	98.4 %

TESTING

- Consisted of Junit testing, Mockito, and Selenium
- 81% Coverage
- Junit
 - Domains, DTO's
- Mockito
 - Controllers, Service
- Tested the whole system against user stories.
 - Example: As a User, I want to **create a list** so that I can add items.

SONARQUBE

Book-Library-Project ☆ master

December 16, 2020, 1:12 PM Version 0.0.1-SNAPSHOT

Overview Issues Security Hotspots Measures Code Activity

Project Settings ▾ Project Information

QUALITY GATE STATUS ⓘ

Passed

All conditions passed.

MEASURES

New Code

Overall Code

0

Bugs

Reliability

A

2

Vulnerabilities

Security

D

0

Security Hotspots ⓘ

100%

Reviewed

Security Review

A

1d 3h

Debt

100

Code Smells

Maintainability

A

0.0%

Coverage on 80 Lines to cover

Unit Tests

0.0%

Duplications on 435 Lines

Duplicated Blocks







0

SONARQUBE

- Shows vulnerabilities in code
- 2 Vulnerabilities were the create mapping in the controllers.
- Entities should not be used as arguments of "@RequestMapping" methods
- Tried to fix but required mass refactoring of code, (both front and back-end) did not have time

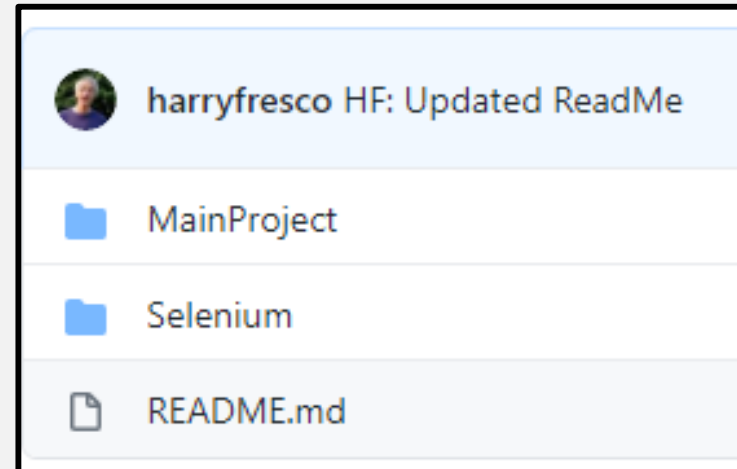
```
// Create method
@PostMapping("/create")
public ResponseEntity<FullListDTO> create(@RequestBody FullList fulllist) {
```

Replace this persistent entity with a simple POJO or DTO object. Why is this an issue?

 Vulnerability   Critical  Open  Not assigned  10min effort [Comment](#)

SELENIUM

- Used Selenium to perform user-acceptance testing.
- Very intuitive
- Was only able to implement two methods:
 - Create List
 - Create Item in list
- In it's own project.
- Needs MainProject running



DEMO

- To demonstrate:
- List – **CRUD**
- Item – **CRUD** – and mark as 'Done'

SPRINT I

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (7)
HFTDLP-5	As a User, I want to be able to create a List on the system through a webpage so that I can later add list-items to it.	Story	Highest	DONE	4
HFTDLP-8	As a User, I want to be able to read a List on the system through a webpage so that I can later add list-items to it.	Story	Highest	DONE	2
HFTDLP-14	As a User, I want to be able to delete a List on the system through a webpage so that I can make sure the list's info is correct.	Story	High	DONE	1



- Was to do with lists and their CRUD functions (without update)
- Took a while to get a strong back-end but after that it went easier
- Updating Item was prolonged until a later sprint because from previous project, updating items took longer than predicted. (MVP)
- Used Postman to test first and then created a very basic HTML page
- Used H2 to host local database while testing

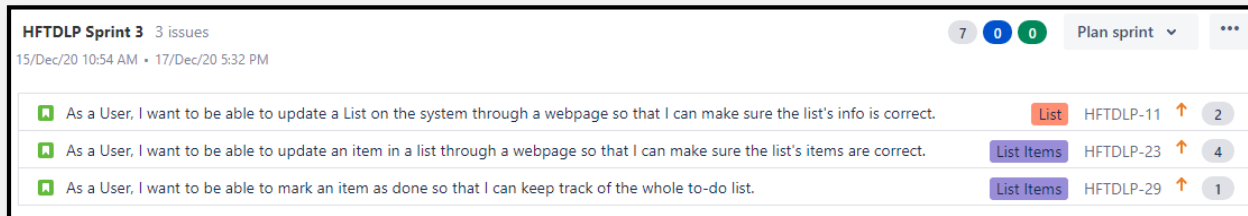
SPRINT 2

Completed Issues							View in Issue Navigator
Key	Summary	Issue Type	Priority	Status	Story Points (7)		
HFTDLP-17	As a User, I want to be able to create an item in a list on the system through a webpage so that I can later view lists and their items.	Story	Highest	DONE	4		
HFTDLP-20	As a User, I want to be able to read an item in a list through a webpage so that I can later mark it as donet.	Story	Highest	DONE	2		
HFTDLP-26	As a User, I want to be able to delete an item in a list through a webpage so that I can make sure the list is correct.	Story	High	DONE	1		



- Was to do with items in lists and their CRUD functions (without update)
- As list functionality was already created, it was a very easy sprint to complete
- This is where I knew this project would go easier than expected.
- Updating lists was also prolonged until the next sprint.

SPRINT 3



- Was to do with items and lists
- Had a few issues with CORS
- Once Lists can be updated, this made Items be updated easier.
- Rest of the sprint went very quickly.
- The marking of an item as done was very easy, as it required changing a bool value and adding a strikethrough on the HTML text.

CONCLUSION

- **What went well**

- Database design and implementation
- Using Jira again, helped layout the project
- Adding and deleting items from Lists
- Back-end was implemented quickly, left time to neaten front-end

- **What I enjoyed**

- Creating and neatening the HTML webpage
- Finding a way to show that an item was marked as done
- Selenium was very intuitive

- **What could be improved**

- An even more intuitive way to display the lists and their items
- Refresh the page when adding items
- Remove the vulnerabilities shown in SonarQube
- More Selenium Tests
- JavaScript could have been laid out better, more separate functions

- **What have I taken away / is useful for future**

- Always start by testing with Postman and then create a very basic front-end before implementing things like Bootstrap
- Using Jira is very useful to get a project started