Creating a To-do List

David Barringer

Introduction

The team:

- David Barringer - QA Academy Trainee

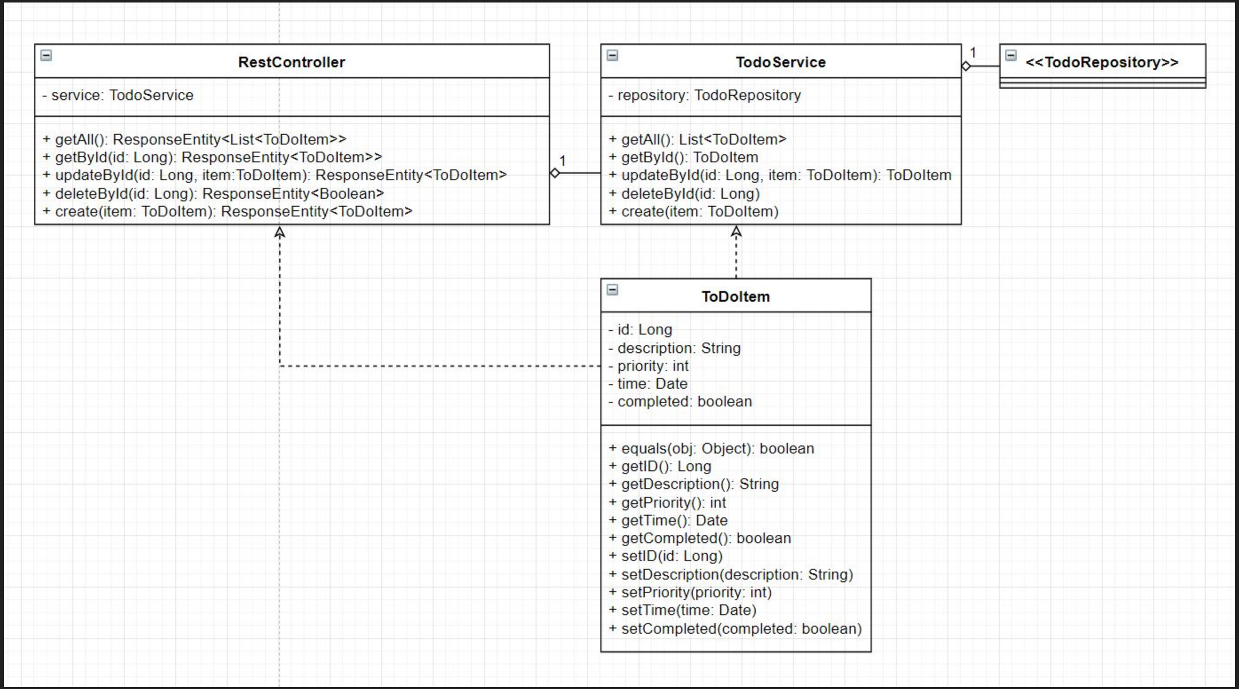
The task:

- Create a to-do list
- CRUD of items in list
- Front-end using HTML, CSS, JS
- Back-end using Java, Spring Boot
- Data stored in MySql database

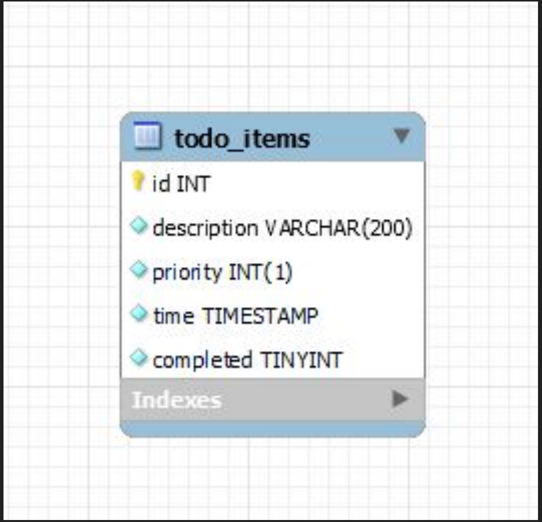
Requirement Analysis

- There’s no requirements for multiple users/lists
- Single list => single page
- UI should be simple
- Web page should be responsive
    - Create page independent of back-end
    - Create back-end

UML and ERD

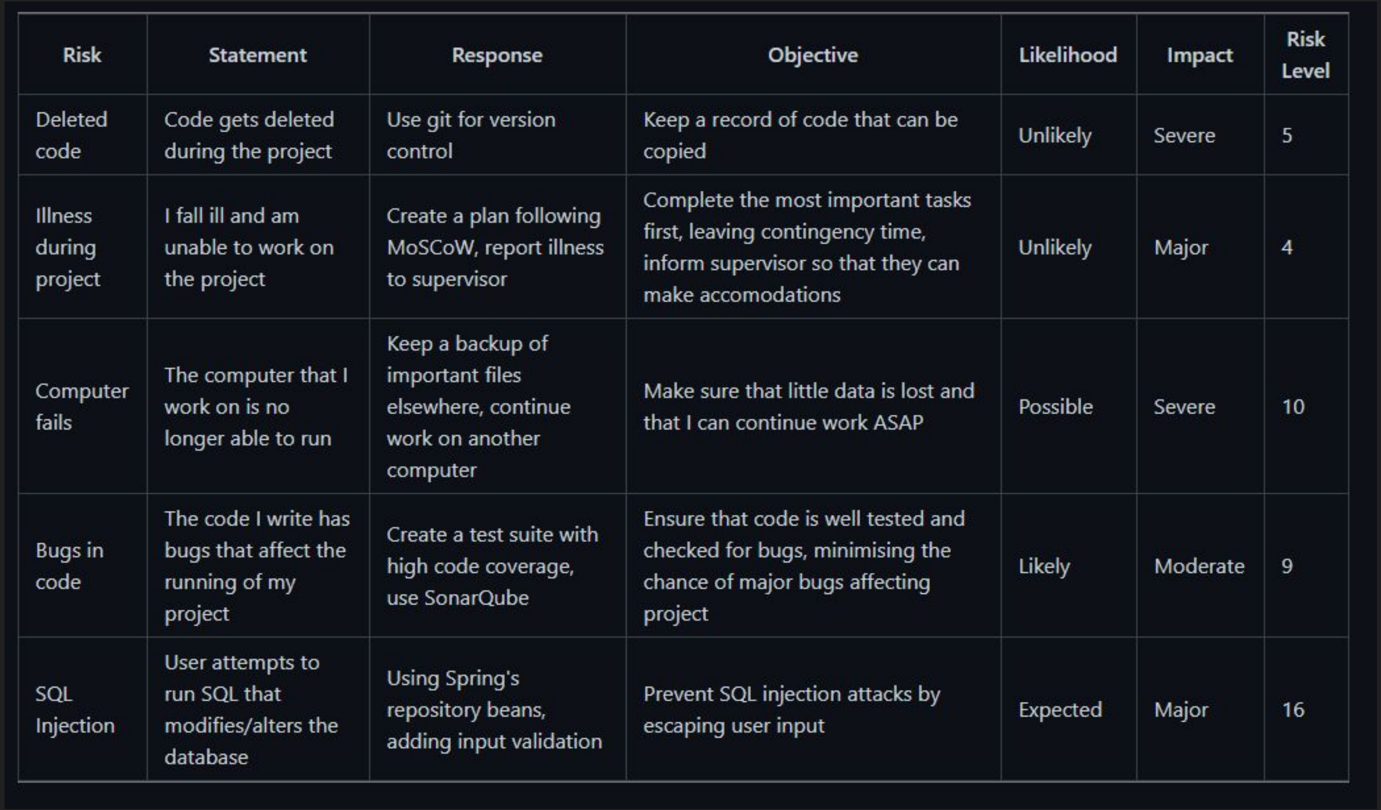


RestController
- service: TodoService
+ getAll(): ResponseEntity<List<ToDoItem>>
+ getById(id: Long): ResponseEntity<ToDoItem>>
+ updateById(id: Long, item:ToDoItem): ResponseEntity<ToDoItem>
+ deleteById(id: Long): ResponseEntity<Boolean>
+ create(item: ToDoItem): ResponseEntity<ToDoItem>
1
TodoService
- repository: TodoRepository
+ getAll(): List<ToDoItem>
+ getById(): ToDoItem
+ updateById(id: Long, item: ToDoItem): ToDoItem
+ deleteById(id: Long)
+ create(item: ToDoItem)
1
<<TodoRepository>>
ToDoItem
- id: Long
- description: String
- priority: int
- time: Date
- completed: boolean
+ equals(obj: Object): boolean
+ getID(): Long
+ getDescription(): String
+ getPriority(): int
+ getTime(): Date
+ getCompleted(): boolean
+ setID(id: Long)
+ setDescription(description: String)
+ setPriority(priority: int)
+ setTime(time: Date)
+ setCompleted(completed: boolean)



todo_items
id INT
description VARCHAR(200)
priority INT(1)
time TIMESTAMP
completed TINYINT
Indexes

Risk Assessment



Risk	Statement	Response	Objective	Likelihood	Impact	Risk Level
Deleted code	Code gets deleted during the project	Use git for version control	Keep a record of code that can be copied	Unlikely	Severe	5
Illness during project	I fall ill and am unable to work on the project	Create a plan following MoSCoW, report illness to supervisor	Complete the most important tasks first, leaving contingency time, inform supervisor so that they can make accomodations	Unlikely	Major	4
Computer fails	The computer that I work on is no longer able to run	Keep a backup of important files elsewhere, continue work on another computer	Make sure that little data is lost and that I can continue work ASAP	Possible	Severe	10
Bugs in code	The code I write has bugs that affect the running of my project	Create a test suite with high code coverage, use SonarQube	Ensure that code is well tested and checked for bugs, minimising the chance of major bugs affecting project	Likely	Moderate	9
SQL Injection	User attempts to run SQL that modifies/alters the database	Using Spring's repository beans, adding input validation	Prevent SQL injection attacks by escaping user input	Expected	Major	16

Technologies Used

- Management - Jira
- Version Control - git
- Application - Spring Boot
- Testing - JUnit5, Mockito, Spring Boot Test, Selenium
- Code review - SonarQube
- Building - Maven

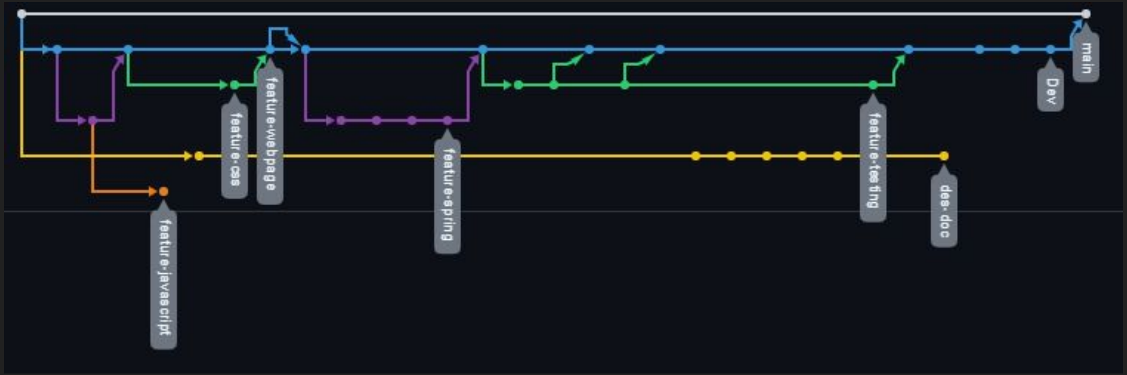
CI - Version Control

Project developed using feature-branch model

Front-end, back-end and tests written on separate branches

Then refactored and merged to dev branch

Working version passing all tests merged to main



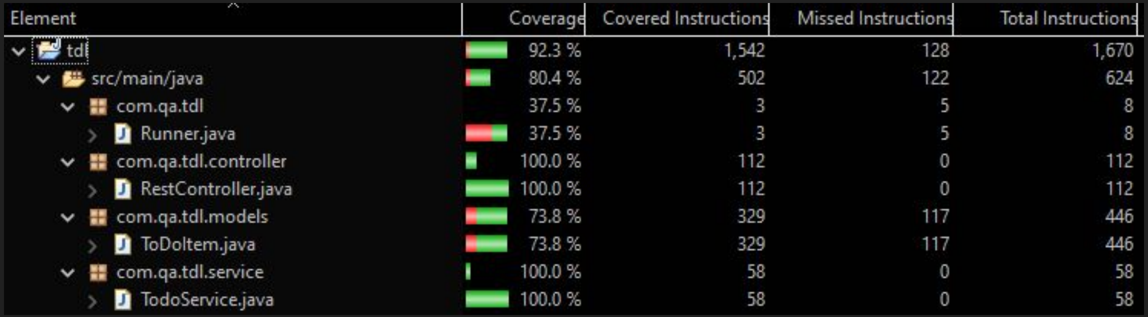
Testing

Unit tests for business logic: JUnit5, Mockito

Integration tests for Spring: Spring Boot Test, JUnit5, MockMVC

User Acceptance Testing: Selenium, Spring Boot Test

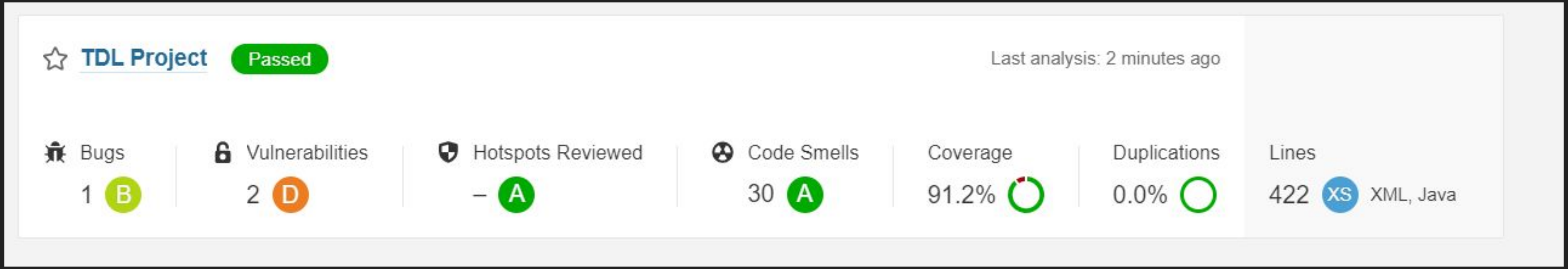
Achieved 80.5% coverage



Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
tdl	92.3 %	1,542	128	1,670
src/main/java	80.4 %	502	122	624
com.qa.tdl	37.5 %	3	5	8
Runner.java	37.5 %	3	5	8
com.qa.tdl.controller	100.0 %	112	0	112
RestController.java	100.0 %	112	0	112
com.qa.tdl.models	73.8 %	329	117	446
ToDoItem.java	73.8 %	329	117	446
com.qa.tdl.service	100.0 %	58	0	58
TodoService.java	100.0 %	58	0	58

Code Review

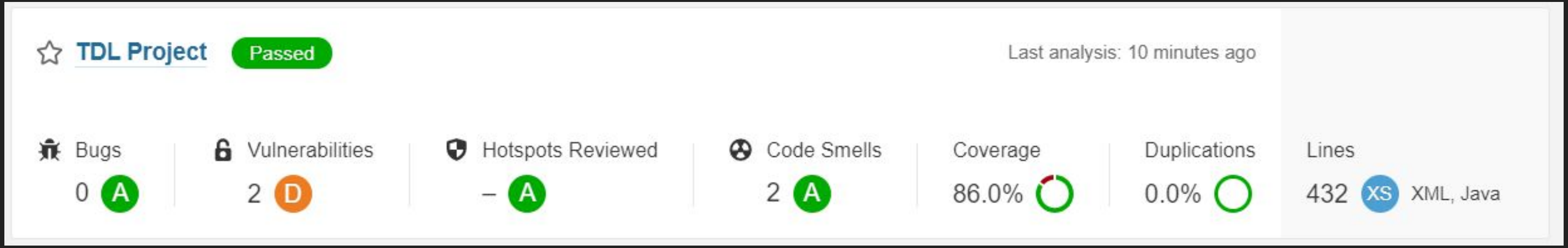
Used SonarQube



TDL Project Passed

Last analysis: 2 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
1 B	2 D	– A	30 A	91.2%	0.0%	422 XS XML, Java

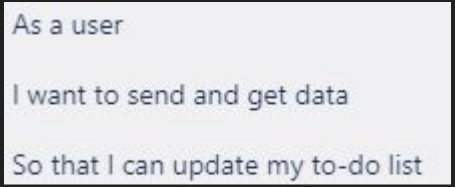


TDL Project Passed

Last analysis: 10 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	2 D	– A	2 A	86.0%	0.0%	432 XS XML, Java

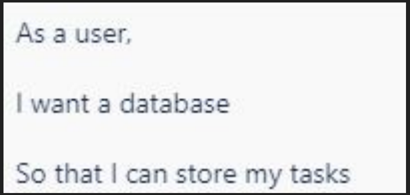
Demonstration



As a user

I want to send and get data

So that I can update my to-do list



As a user,

I want a database

So that I can store my tasks

Sprint Review

What was completed

- Planning: UML, ERD, Risk Assessment, Jira board
- Working front-end using HTML, CSS and JS
- Requests handled by Spring Boot
- Data stored correctly in DB

What wasn’t completed

- MVP achieved
- Further development on front-end would be useful

Sprint Retrospective

- Completed all tasks initially set
- Project better managed than previous
- Story points for writing tests were underestimated
- Some blockers that were eventually fixed/avoided (CORS, EqualityVerifier, Selenium with Spring Boot)
- Progress during documentation slower

Conclusion

Project meets MVP requirements

Further development:

- Filtering/sorting items
- Better & more responsive styling
- Adding “urgency” to uncompleted tasks

Questions