

To-Do List Application

Sam Costello

20DEC SOFTWARE1

Introduction

- Name : Sam Costello
- How did I approach specification?
- Approach: I wanted to make sure that my crud functionality for all aspects of the application was working before going on to working on anything extra

Concept

- Research
- Created ERD
- Created Risk Management Log
- Used Jira
- Set up a daily notepad

Sprint Plan

- What: Applications CRUD functionality
- How: Back end -> Front end
- Input: Product Backlog
- Output: fully working to do list

Consultant Journey

- Version control: git
 - Programme language: javascript - (HTML, CSS) - Java
 - Database: local-H2
 - Testing: - unit - Junit
 - intergration
 - acceptance – selenium
- Static analysis - sonarqube
- Jira

CI

Try to commit as much as possible in a rolling manner

Reduce merge conflicts

Feature branch model

Maven build tool

Junit

Testing

- Unit Testing
- Integration Testing
- User Acceptance Testing

```
task = new TaskDomain(1L, "Hoover", 2, "01-01-01", "Hoover downstairs", false, null);

Assertions.assertThat(task).isNotNull();
task.setTaskId(2L);
task.setAct("Hoover");
task.setPriority(2);
task.setDate("22-02-2020");
task.setNotes("testing");
task.setCompleted(true);
task.setToDo(null);
Assertions.assertThat(task.getTaskId()).isNotNull();
Assertions.assertThat(task.getAct()).isNotNull();
Assertions.assertThat(task.getPriority()).isNotNull();
Assertions.assertThat(task.getDate()).isNotNull();
Assertions.assertThat(task.getNotes()).isNotNull();
```

Unit testing

The Unit tests


```

@Test
public void readById() throws Exception{

    TaskDTO expectedResult1 = new TaskDTO(1L, "Practice", 3, "22/06/2022", "this is a test to make sure we can prepopulate the database", false);
    //set it up
    MockHttpServletRequestBuilder mockRequest = MockMvcRequestBuilders.request(HttpMethod.GET, "http://localhost:8080/task/read/" + ID);
    //set the expectations
    ResultMatcher matchStatus = MockMvcResultMatchers.status().isOk();
    ResultMatcher matchContent = MockMvcResultMatchers.content().json(jsonifier.writeValueAsString(expectedResult1));
    //perform the actions

    this.mock.perform(mockRequest).andExpect(matchStatus).andExpect(matchContent);
}

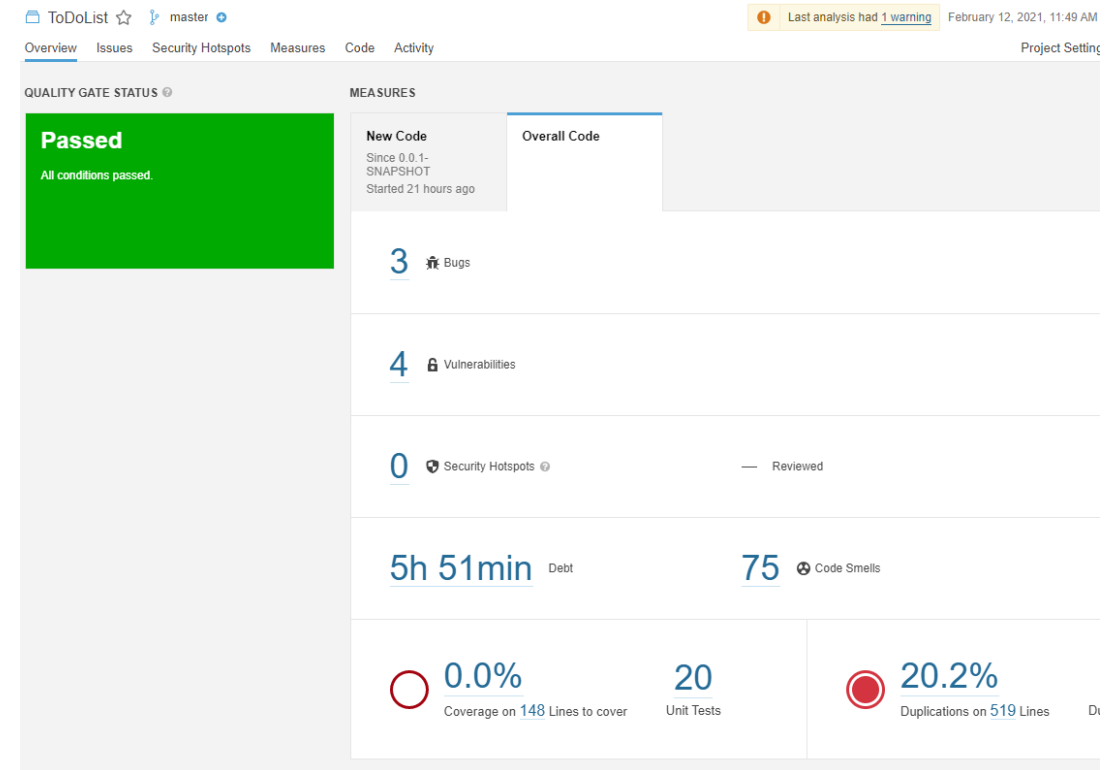
```

- We took individual classes – controller and service, and combined them to test them as a group
- Compliance
- CRUD

Integration testing

Static analysis - Sonarqube

- Continuous inspection
- Reduces errors, alerts them to you



User Acceptance – selenium

- Behaviour Driven Development
- //given
- //when
- //then

```
    readByIdTest() {  
        report.startTest("Read Individual Task Test");  
  
        //given the url to the task.html page  
        driver.get(URL);  
  
        //when i click the read button  
        targ = driver.findElement(By.xpath("/html/body/button[3]"));  
        targ.click();  
  
        //and pass in an id  
        targ = driver.findElement(By.id("readbyid"));  
        targ.sendKeys("1");  
        targ = driver.findElement(By.xpath("//*[@id=\"readbutton\"]/div/"));  
        targ.click();  
  
        //then i should see the task appear on screen  
        targ = new WebDriverWait(driver, 5).until(ExpectedConditions.presenceOfElementLocated(By.id("taskId")));  
        String result = targ.getText();  
        String expected = "{\\\"taskId\\\":1,\"";  
  
        //test logging if pass or fail  
        if(result.contains(expected)) {  
            test.log(LogStatus.PASS, expected);  
        }else {  
            test.log(LogStatus.FAIL, "failed to read");  
        }  
  
        //assert that what comes back contains the char sequence provide  
        Assert.assertTrue(result.contains(expected));  
    }
```

Extent Reports

TESTS
<div><div></div><div></div><div></div><div></div><div></div></div>
Read All Task Test Pass
Update Task Test Pass
Read Individual Task Test Pass
Create Task Test Pass
Delete Task Test Pass

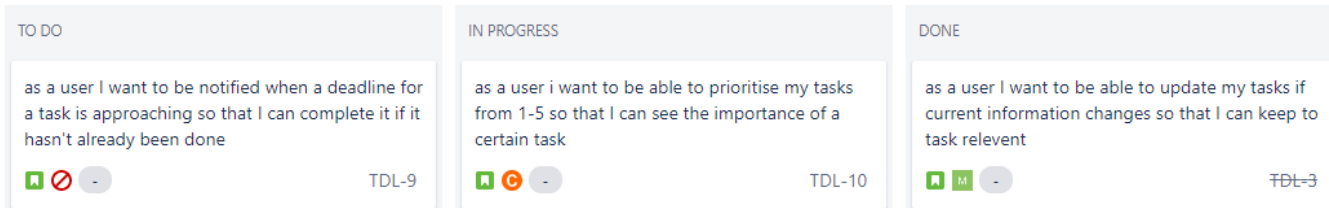
Read All Task Test

2021-02-12 09:40:22 2021-02-12 09:40:24 0h 0m 1s+632ms

STATUS	TIMESTAMP	DETAILS
✓	09:40:24	2 Hoover need to hoover all of downstairs

- Refreshing GUI

Sprint Review



- Completion of MVP
- MoSCoW on user stories (W, C, M)
- What got let behind
- Notification idea
- Ordering the priority
- Collaboration S

Sprint Retrospective

- What went well?
 - Flexibility
 - Focus
 - Simple plan
-
- What could be improved
 - More detailed plan
 - Lombok
 - POM design pattern for selenium
 - HTML updating table



Demonstration

```
4th
-tasks complete
5th
- to do complete
-some testing complete
6th+7th
-weekend
8th
- crud for task js (didnt finish update)
9th
- crud for to do js (again update not done)
10th
- HTML print all results from crud to screen
-start on controller testing
11th
- selenium acceptatnce testing plus reports for task
12th
-update the project diagrams and read me
-finish selenium to do - (i've had to disable the delete test - explain in pres)
13th+14th
-weekend
15th
-
16th
- presentation day
|
```

- Reflections on project
- Linking a front end form to a back end database and enabling to pull information from that database and display it on the page

Conclusion



Questions