Practical Project: Hobby Web Application (HWA)

British Serial Killer database

By Phoebe Stirzaker

Introduction

- Approaching the project

⇨ Coming up with a database idea.

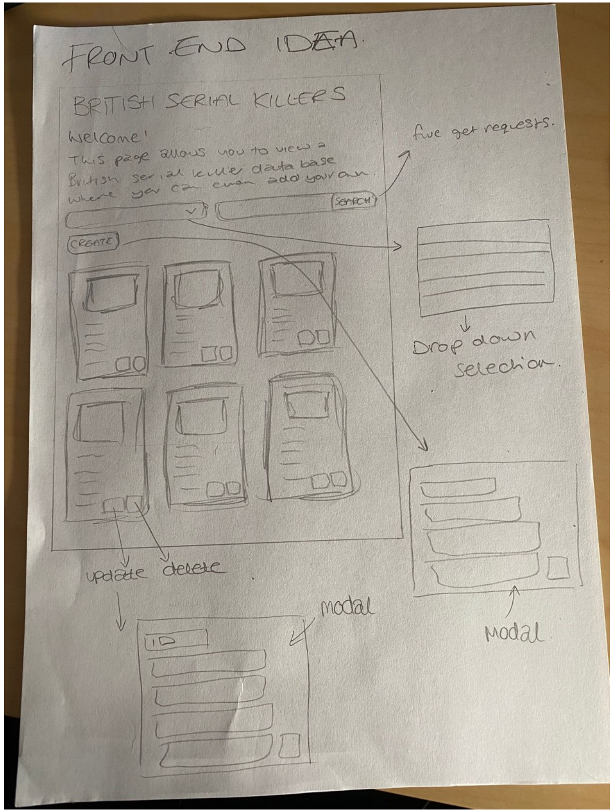
✔ Serial Killers.

⇨ Reading the deliverables thoroughly and coming up with a plan of action.

⇨ Drawing up a front end plan.

⇨ Creating a break down of all elements.

✔ Jira.



Welcome!
This page allows you to view a
British serial killer data base
where you can even add your own.

Concept

- What did I want my database to do?

⇨ Store British serial killers.

⇨ CRUD targets:

✔ Create

✔ (Read) Get all/Get by ID:

✰ Get by first name

✰ Get by last name

✰ Get by place

✰ Get by confirmed kills

✰ Get by zodiac sign

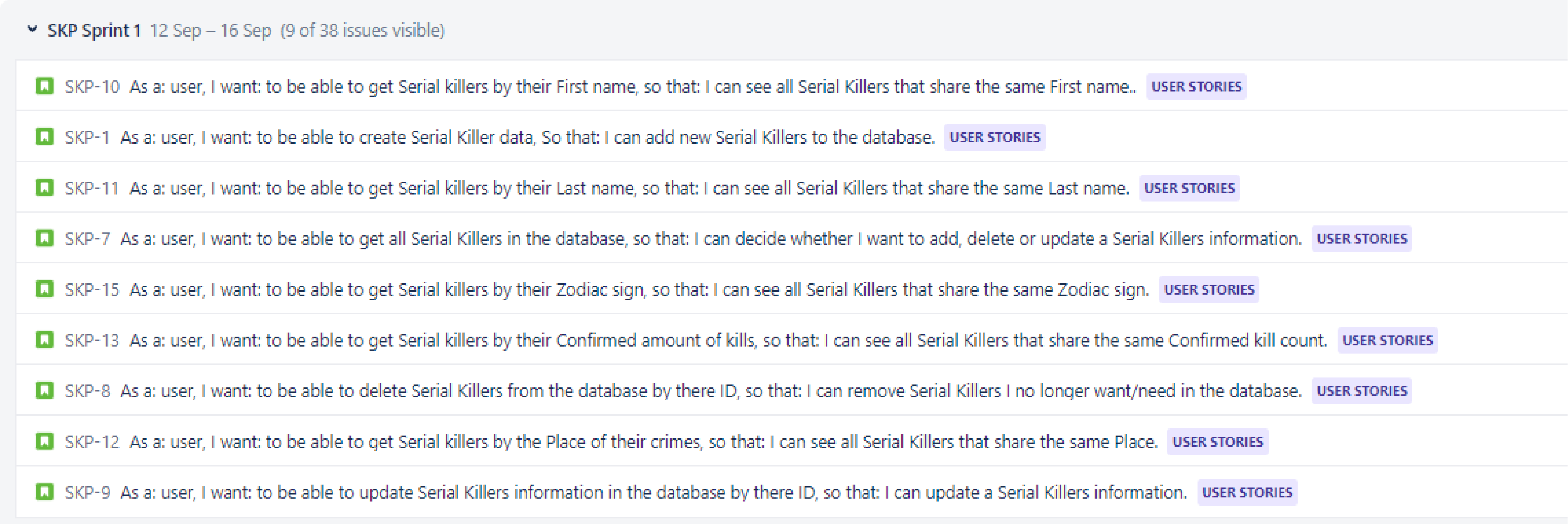
✔ Update

✔ Delete

Sprint plan

- What needed to be included?

⇨ User stories



SKP Sprint 1 12 Sep – 16 Sep (9 of 38 issues visible)

SKP-10 As a: user, I want: to be able to get Serial killers by their First name, so that: I can see all Serial Killers that share the same First name.. USER STORIES

SKP-1 As a: user, I want: to be able to create Serial Killer data, So that: I can add new Serial Killers to the database. USER STORIES

SKP-11 As a: user, I want: to be able to get Serial killers by their Last name, so that: I can see all Serial Killers that share the same Last name. USER STORIES

SKP-7 As a: user, I want: to be able to get all Serial Killers in the database, so that: I can decide whether I want to add, delete or update a Serial Killers information. USER STORIES

SKP-15 As a: user, I want: to be able to get Serial killers by their Zodiac sign, so that: I can see all Serial Killers that share the same Zodiac sign. USER STORIES

SKP-13 As a: user, I want: to be able to get Serial killers by their Confirmed amount of kills, so that: I can see all Serial Killers that share the same Confirmed kill count. USER STORIES

SKP-8 As a: user, I want: to be able to delete Serial Killers from the database by there ID, so that: I can remove Serial Killers I no longer want/need in the database. USER STORIES

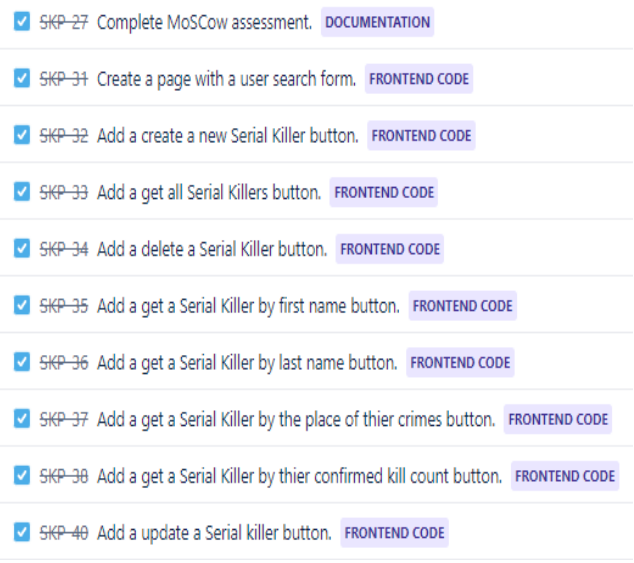
SKP-12 As a: user, I want: to be able to get Serial killers by the Place of their crimes, so that: I can see all Serial Killers that share the same Place. USER STORIES

SKP-9 As a: user, I want: to be able to update Serial Killers information in the database by there ID, so that: I can update a Serial Killers information. USER STORIES

Sprint plan

- What needed to be included?

⇨ Tasks



☑ ~~SKP-27~~ Complete MoSCow assessment. DOCUMENTATION

☑ ~~SKP-31~~ Create a page with a user search form. FRONTEND CODE

☑ ~~SKP-32~~ Add a create a new Serial Killer button. FRONTEND CODE

☑ ~~SKP-33~~ Add a get all Serial Killers button. FRONTEND CODE

☑ ~~SKP-34~~ Add a delete a Serial Killer button. FRONTEND CODE

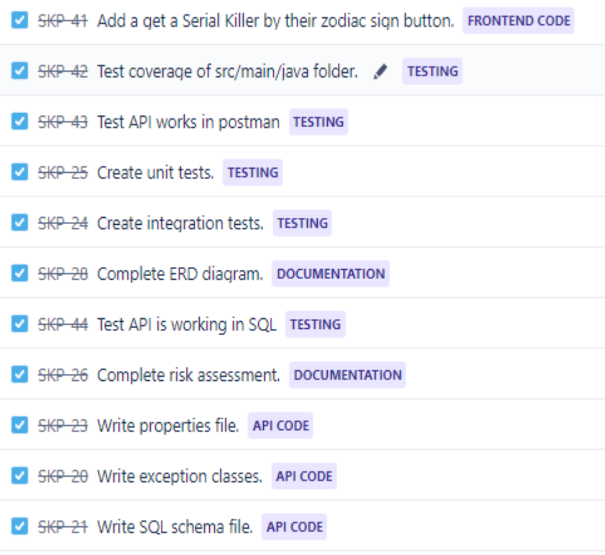
☑ ~~SKP-35~~ Add a get a Serial Killer by first name button. FRONTEND CODE

☑ ~~SKP-36~~ Add a get a Serial Killer by last name button. FRONTEND CODE

☑ ~~SKP-37~~ Add a get a Serial Killer by the place of thier crimes button. FRONTEND CODE

☑ ~~SKP-38~~ Add a get a Serial Killer by thier confirmed kill count button. FRONTEND CODE

☑ ~~SKP-40~~ Add a update a Serial killer button. FRONTEND CODE



☑ ~~SKP-41~~ Add a get a Serial Killer by their zodiac sign button. FRONTEND CODE

☑ ~~SKP-42~~ Test coverage of src/main/java folder. TESTING

☑ ~~SKP-43~~ Test API works in postman TESTING

☑ ~~SKP-25~~ Create unit tests. TESTING

☑ ~~SKP-24~~ Create integration tests. TESTING

☑ ~~SKP-28~~ Complete ERD diagram. DOCUMENTATION

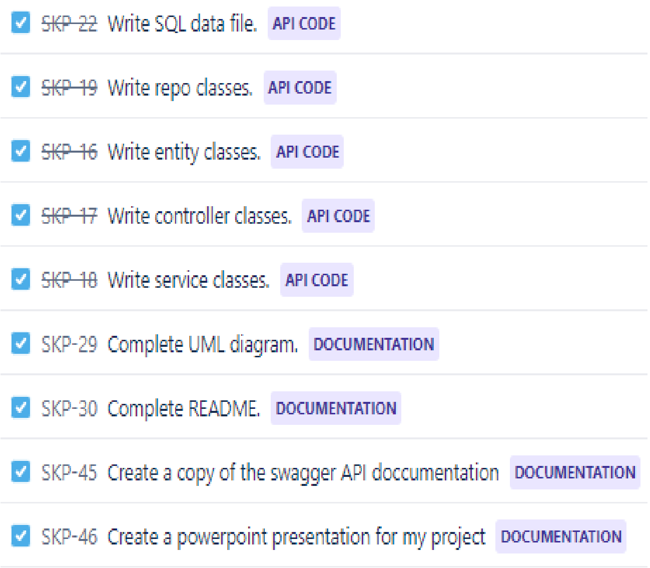
☑ ~~SKP-44~~ Test API is working in SQL TESTING

☑ ~~SKP-26~~ Complete risk assessment. DOCUMENTATION

☑ ~~SKP-23~~ Write properties file. API CODE

☑ ~~SKP-20~~ Write exception classes. API CODE

☑ ~~SKP-21~~ Write SQL schema file. API CODE



☑ ~~SKP-22~~ Write SQL data file. API CODE

☑ ~~SKP-19~~ Write repo classes. API CODE

☑ ~~SKP-16~~ Write entity classes. API CODE

☑ ~~SKP-17~~ Write controller classes. API CODE

☑ ~~SKP-18~~ Write service classes. API CODE

☑ SKP-29 Complete UML diagram. DOCUMENTATION

☑ SKP-30 Complete README. DOCUMENTATION

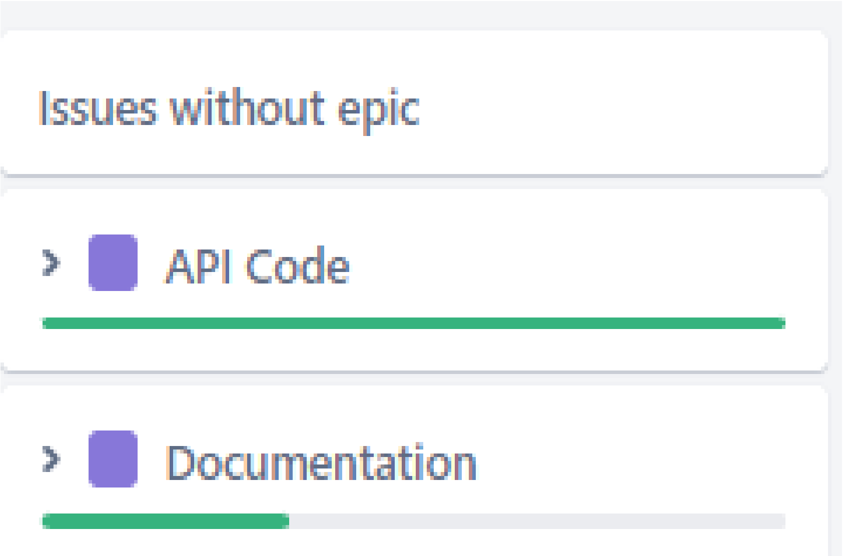
☑ SKP-45 Create a copy of the swagger API doccumentation DOCUMENTATION

☑ SKP-46 Create a powerpoint presentation for my project DOCUMENTATION

Sprint plan

- What needed to be included?

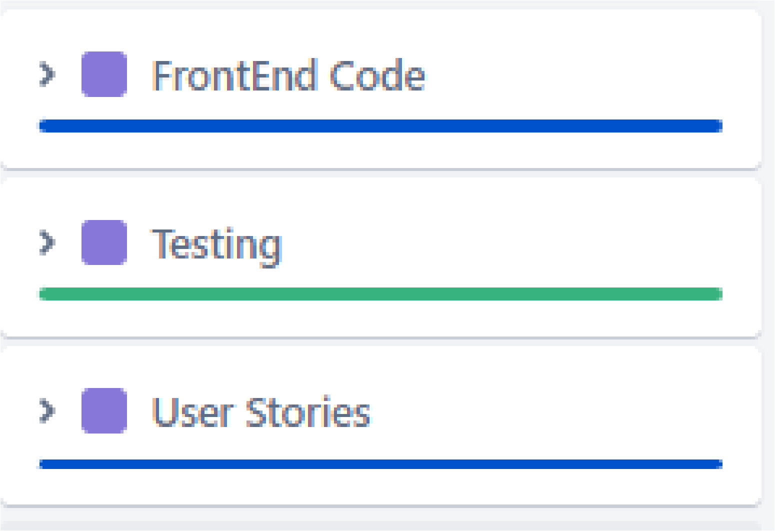
⇨ Epics



Issues without epic

› API Code

› Documentation



› FrontEnd Code

› Testing

› User Stories

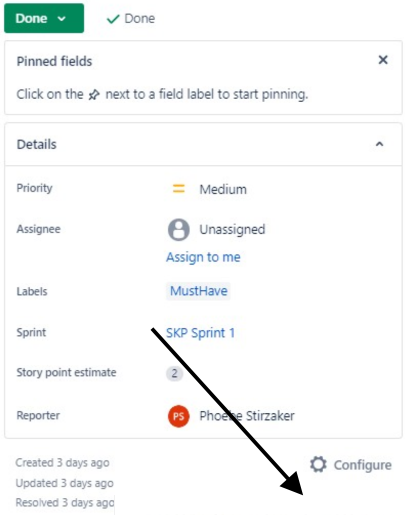
Sprint plan

- What needed to be included?

⇨ Story points

✔ x2 incrementing difficulty

Story point estimate 4



Done ⌄ ✓ Done

Pinned fields ×

Click on the 📌 next to a field label to start pinning.

Details ^

Priority = Medium

Assignee Unassigned

Assign to me

Labels MustHave

Sprint SKP Sprint 1

Story point estimate 2

Reporter PS Phoebe Stirzaker

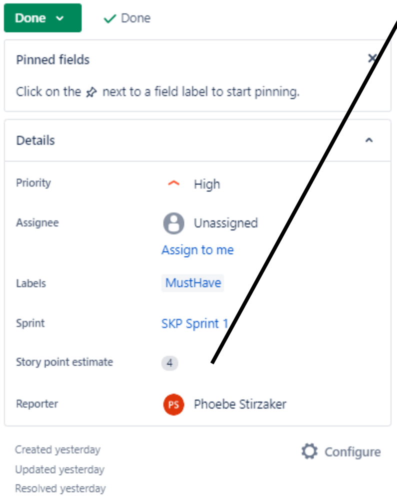
Created 3 days ago

Updated 3 days ago

Resolved 3 days ago

⚙ Configure

Story point estimate 2



Done ⌄ ✓ Done

Pinned fields ×

Click on the 📌 next to a field label to start pinning.

Details ^

Priority ^ High

Assignee Unassigned

Assign to me

Labels MustHave

Sprint SKP Sprint 1

Story point estimate 4

Reporter PS Phoebe Stirzaker

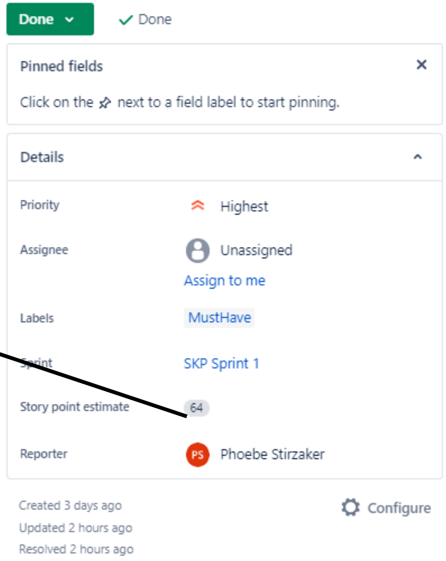
Created yesterday

Updated yesterday

Resolved yesterday

⚙ Configure

Story point estimate 64



Done ⌄ ✓ Done

Pinned fields ×

Click on the 📌 next to a field label to start pinning.

Details ^

Priority Highest

Assignee Unassigned

Assign to me

Labels MustHave

Sprint SKP Sprint 1

Story point estimate 64

Reporter PS Phoebe Stirzaker

Created 3 days ago

Updated 2 hours ago

Resolved 2 hours ago

⚙ Configure

Sprint plan

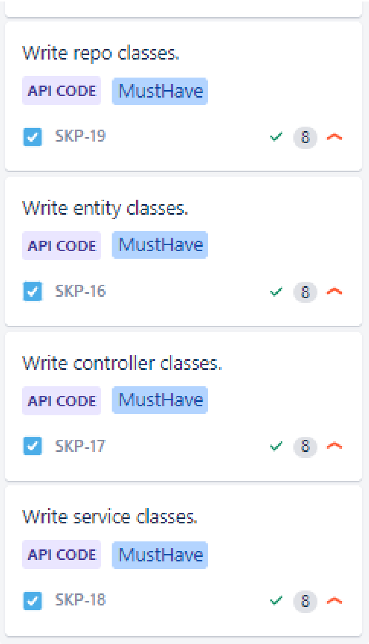
- What needed to be included?

⇨ MoSCoW prioritisation

✔ Must have

✔ Should have

✔ Could have

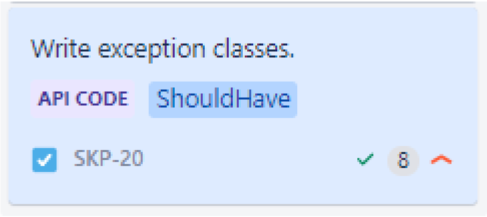


Write repo classes.
API CODE MustHave
☑ SKP-19 ✓ 8 ^

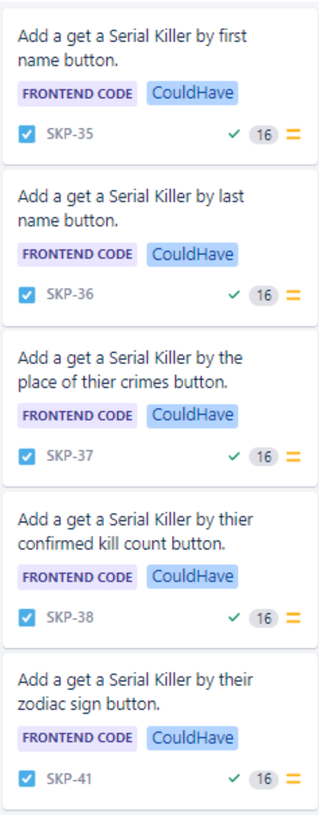
Write entity classes.
API CODE MustHave
☑ SKP-16 ✓ 8 ^

Write controller classes.
API CODE MustHave
☑ SKP-17 ✓ 8 ^

Write service classes.
API CODE MustHave
☑ SKP-18 ✓ 8 ^



Write exception classes.
API CODE ShouldHave
☑ SKP-20 ✓ 8 ^



Add a get a Serial Killer by first name button.
FRONTEND CODE CouldHave
☑ SKP-35 ✓ 16 =

Add a get a Serial Killer by last name button.
FRONTEND CODE CouldHave
☑ SKP-36 ✓ 16 =

Add a get a Serial Killer by the place of thier crimes button.
FRONTEND CODE CouldHave
☑ SKP-37 ✓ 16 =

Add a get a Serial Killer by thier confirmed kill count button.
FRONTEND CODE CouldHave
☑ SKP-38 ✓ 16 =

Add a get a Serial Killer by their zodiac sign button.
FRONTEND CODE CouldHave
☑ SKP-41 ✓ 16 =

Consultant journey

- What technologies have I learned for this project?

⇨ Java

✔ Eclipse IDE (Integrated Development Environment)

⇨ JavaScript

⇨ Bootstrap

⇨ Html

⇨ Google

⇨ Postman

⇨ MySQL

⇨ Git

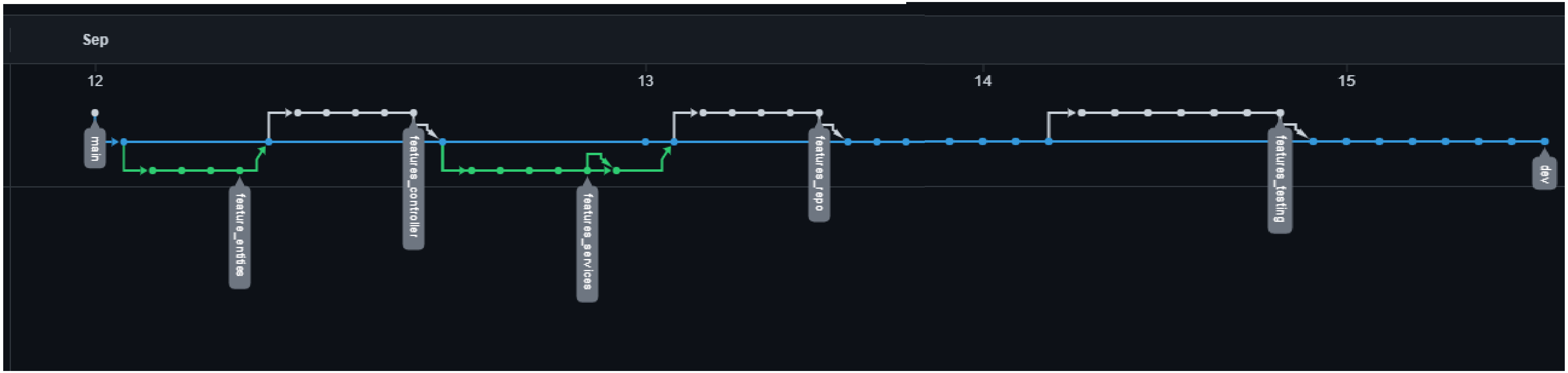
✔ GitHub

✔ Git Bash

Can I help?

Continuous Integration

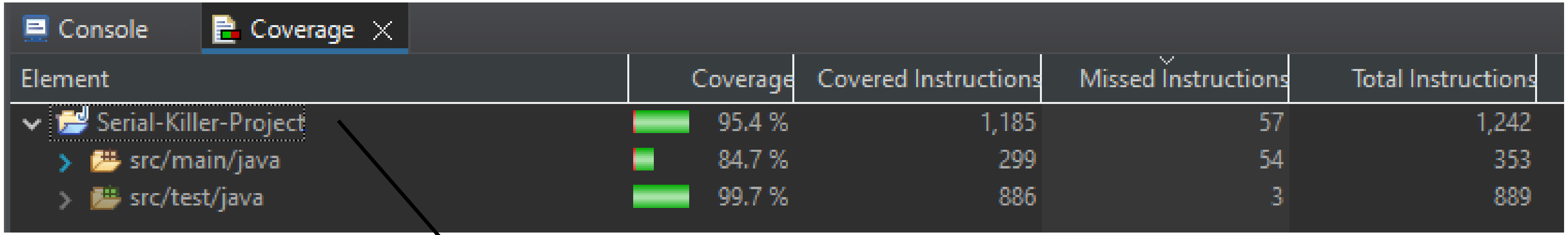
- Using the feature-branch model.



Testing

- What was done for testing?

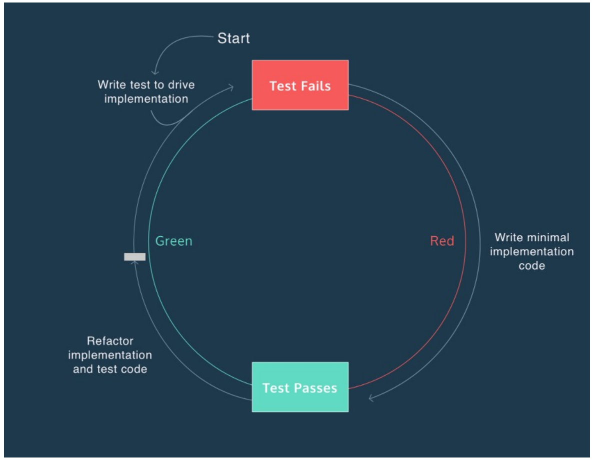
⇨ Coverage



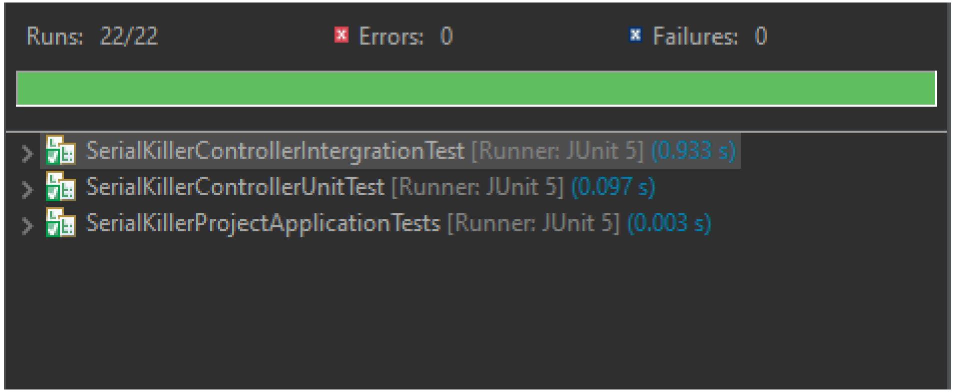
Console | Coverage

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Serial-Killer-Project	95.4 %	1,185	57	1,242
src/main/java	84.7 %	299	54	353
src/test/java	99.7 %	886	3	889

⇨ Red-Green-Refactor



Start
Write test to drive implementation
Test Fails
Red
Write minimal implementation code
Test Passes
Green
Refactor implementation and test code



```
Runs:  22/22          Errors:  0          Failures:  0

> SerialKillerControllerIntergrationTest [Runner: JUnit 5] (0.933 s)
> SerialKillerControllerUnitTest [Runner: JUnit 5] (0.097 s)
> SerialKillerProjectApplicationTests [Runner: JUnit 5] (0.003 s)
```

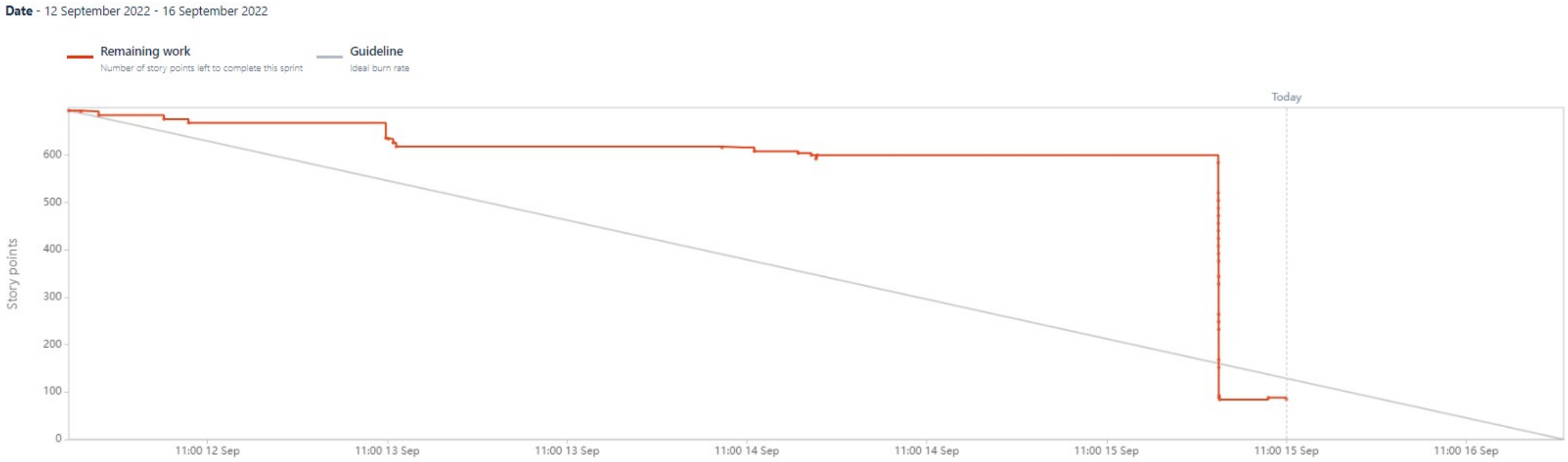
Demonstration

Sprint review

- What did I complete?

⇨ User stories as of Friday 16th September 2022.

⇨ Tasks as of Friday 16th September 2022.



Sprint retrospective

- What went well?

⇨ Understanding issues and fixing them.

⇨ Implementing code through bootstrap.

⇨ Using learnt skills to use in my code.

- What could be improved?

⇨ Time management.

⇨ Testing.

⇨ Styling.

⇨ Knowing when to come back to a problem.

Conclusion

- Overall thoughts and observations.

⇨ The project itself.

✔ It works !

✔ Challenging aspects.

⇨ Future steps.

✔ Manage my time better.

✔ Be more confident in my understanding.

⇨ General notes.

✔ Using google and asking for guidance when facing issues.

✔ Toning down the front end.

Questions?