PRACTICAL PROJECT

Sean Fernyhough

INTRODUCTION

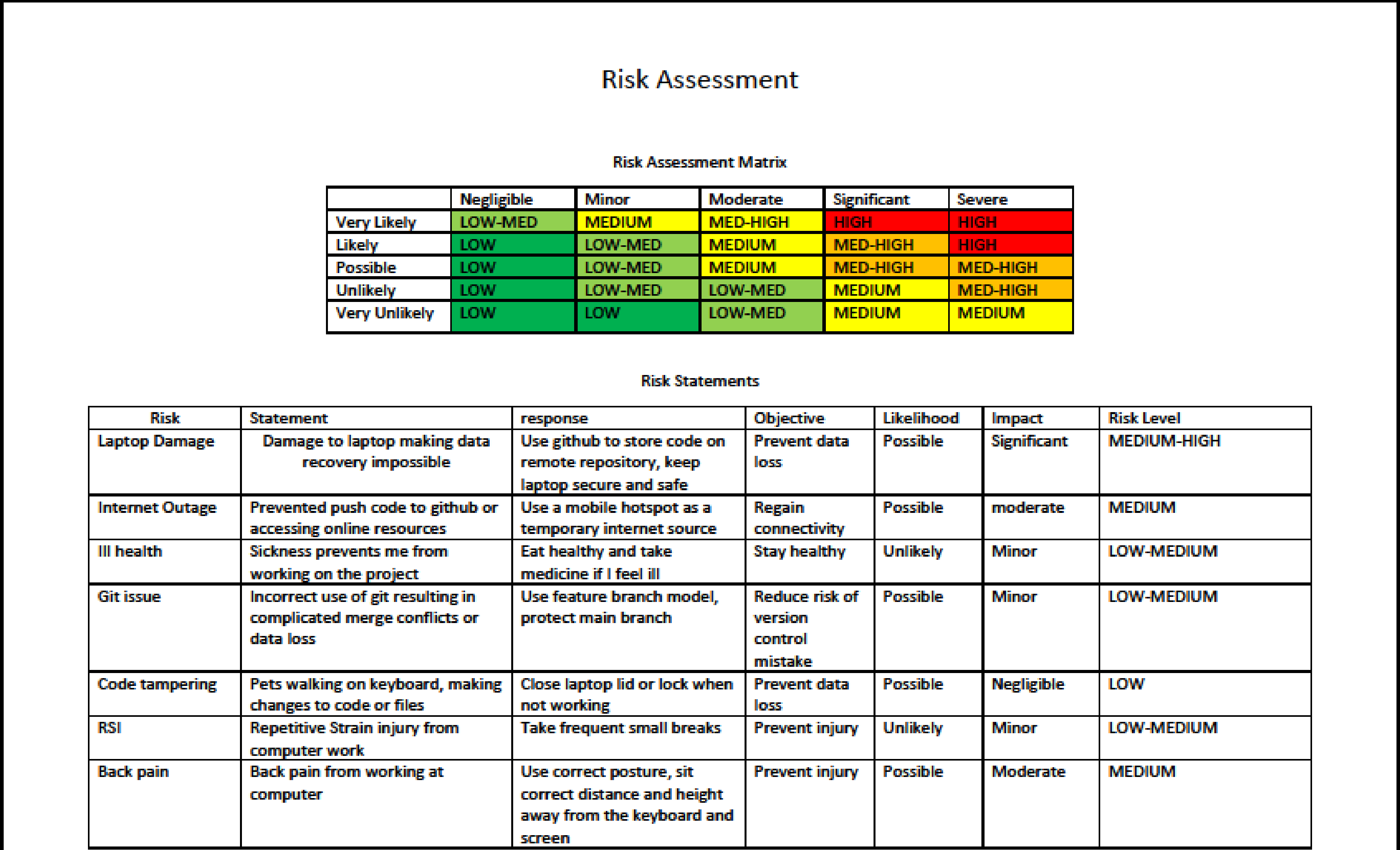
- Movie database web app
  - Two entities
    - Movies
    - Actors
  - Supports CRUD operations for both entities
  - Supports multiple search methods
  - Can fetch a random movie
  - User friendly UI

APPROACH

- Test-driven development
  - Unit test written first

- Planning
  - Risk Assessment
  - ER Diagram
  - UML Diagram

RISK ASSESSMENT



Risk Assessment

Risk Assessment Matrix

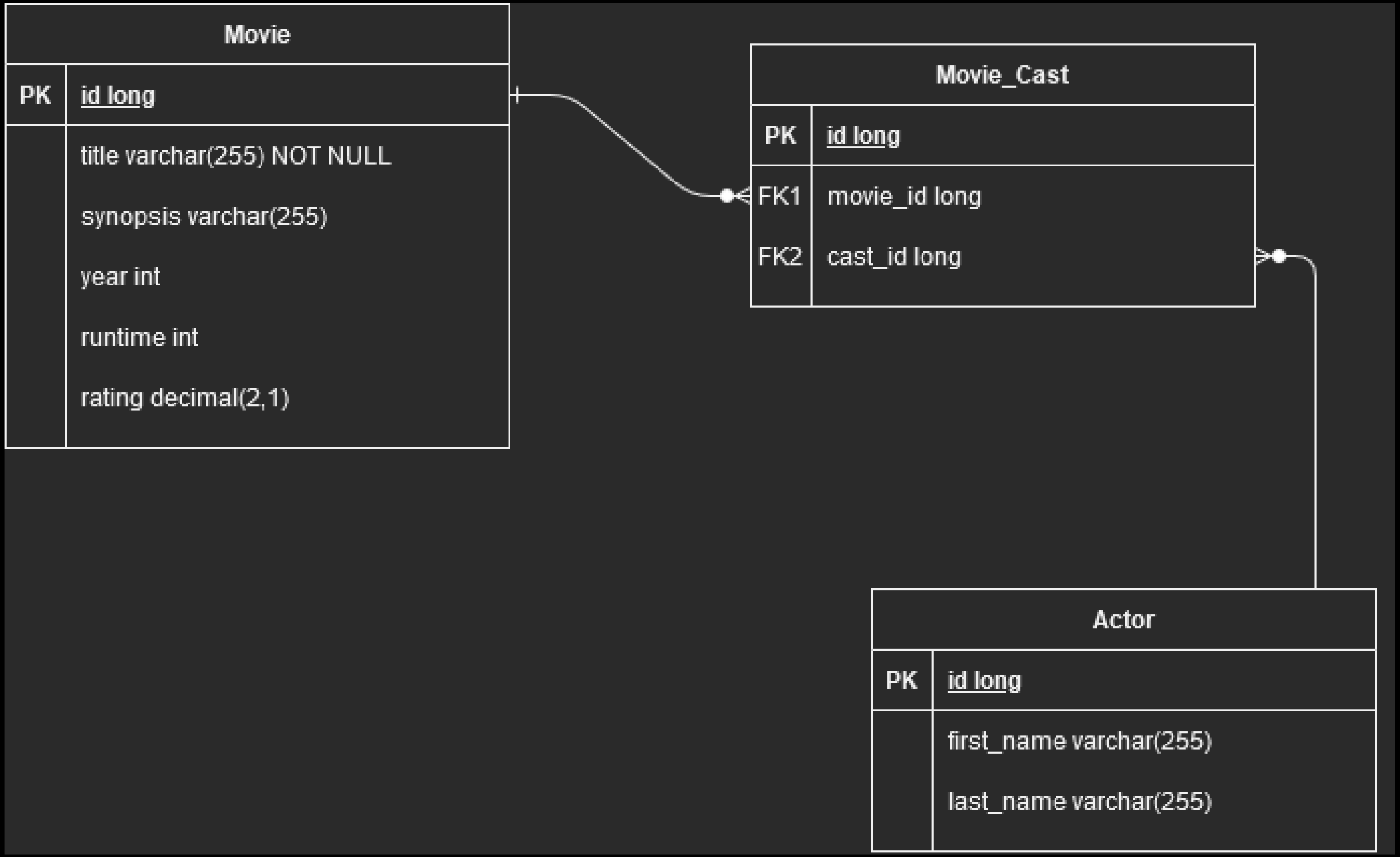
	Negligible	Minor	Moderate	Significant	Severe
Very Likely	LOW-MED	MEDIUM	MED-HIGH	HIGH	HIGH
Likely	LOW	LOW-MED	MEDIUM	MED-HIGH	HIGH
Possible	LOW	LOW-MED	MEDIUM	MED-HIGH	MED-HIGH
Unlikely	LOW	LOW-MED	LOW-MED	MEDIUM	MED-HIGH
Very Unlikely	LOW	LOW	LOW-MED	MEDIUM	MEDIUM

Risk Statements

Risk	Statement	response	Objective	Likelihood	Impact	Risk Level
Laptop Damage	Damage to laptop making data recovery impossible	Use github to store code on remote repository, keep laptop secure and safe	Prevent data loss	Possible	Significant	MEDIUM-HIGH
Internet Outage	Prevented push code to github or accessing online resources	Use a mobile hotspot as a temporary internet source	Regain connectivity	Possible	moderate	MEDIUM
Ill health	Sickness prevents me from working on the project	Eat healthy and take medicine if I feel ill	Stay healthy	Unlikely	Minor	LOW-MEDIUM
Git issue	Incorrect use of git resulting in complicated merge conflicts or data loss	Use feature branch model, protect main branch	Reduce risk of version control mistake	Possible	Minor	LOW-MEDIUM
Code tampering	Pets walking on keyboard, making changes to code or files	Close laptop lid or lock when not working	Prevent data loss	Possible	Negligible	LOW
RSI	Repetitive Strain injury from computer work	Take frequent small breaks	Prevent injury	Unlikely	Minor	LOW-MEDIUM
Back pain	Back pain from working at computer	Use correct posture, sit correct distance and height away from the keyboard and screen	Prevent injury	Possible	Moderate	MEDIUM

ER DIAGRAM

Planning



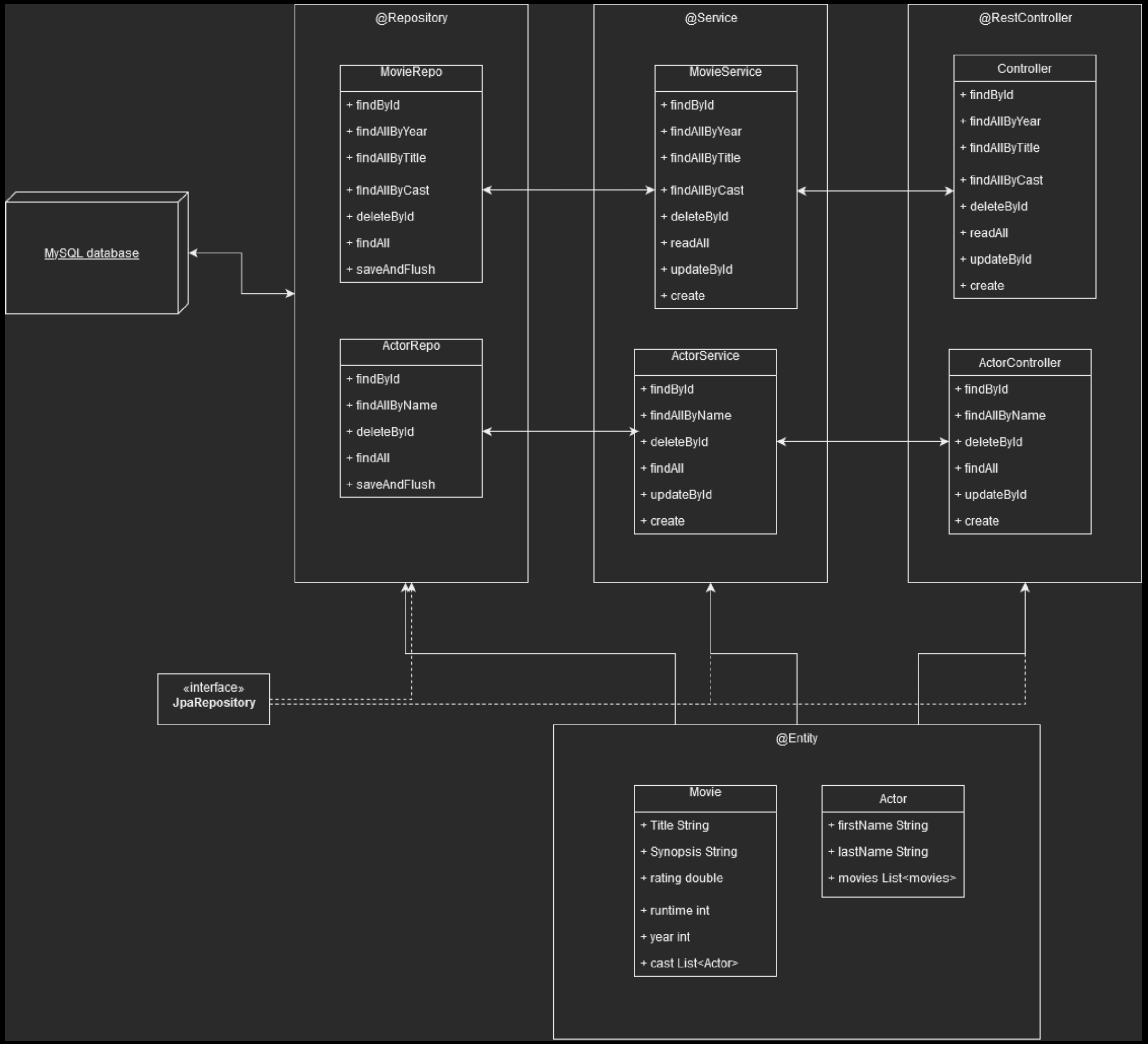
Movie	
PK	id long
	title varchar(255) NOT NULL
	synopsis varchar(255)
	year int
	runtime int
	rating decimal(2,1)

Movie_Cast	
PK	id long
FK1	movie_id long
FK2	cast_id long

Actor	
PK	id long
	first_name varchar(255)
	last_name varchar(255)

UML DIAGRAM

Planning



@Repository

MovieRepo
+ findById
+ findAllByYear
+ findAllByTitle
+ findAllByCast
+ deleteById
+ findAll
+ saveAndFlush

ActorRepo
+ findById
+ findAllByName
+ deleteById
+ findAll
+ saveAndFlush

@Service

MovieService
+ findById
+ findAllByYear
+ findAllByTitle
+ findAllByCast
+ deleteById
+ readAll
+ updateById
+ create

ActorService
+ findById
+ findAllByName
+ deleteById
+ findAll
+ updateById
+ create

@RestController

Controller
+ findById
+ findAllByYear
+ findAllByTitle
+ findAllByCast
+ deleteById
+ readAll
+ updateById
+ create

ActorController
+ findById
+ findAllByName
+ deleteById
+ findAll
+ updateById
+ create

MySQL database

«interface»
JpaRepository

@Entity

Movie
+ Title String
+ Synopsis String
+ rating double
+ runtime int
+ year int
+ cast List<Actor>

Actor
+ firstName String
+ lastName String
+ movies List<movies>

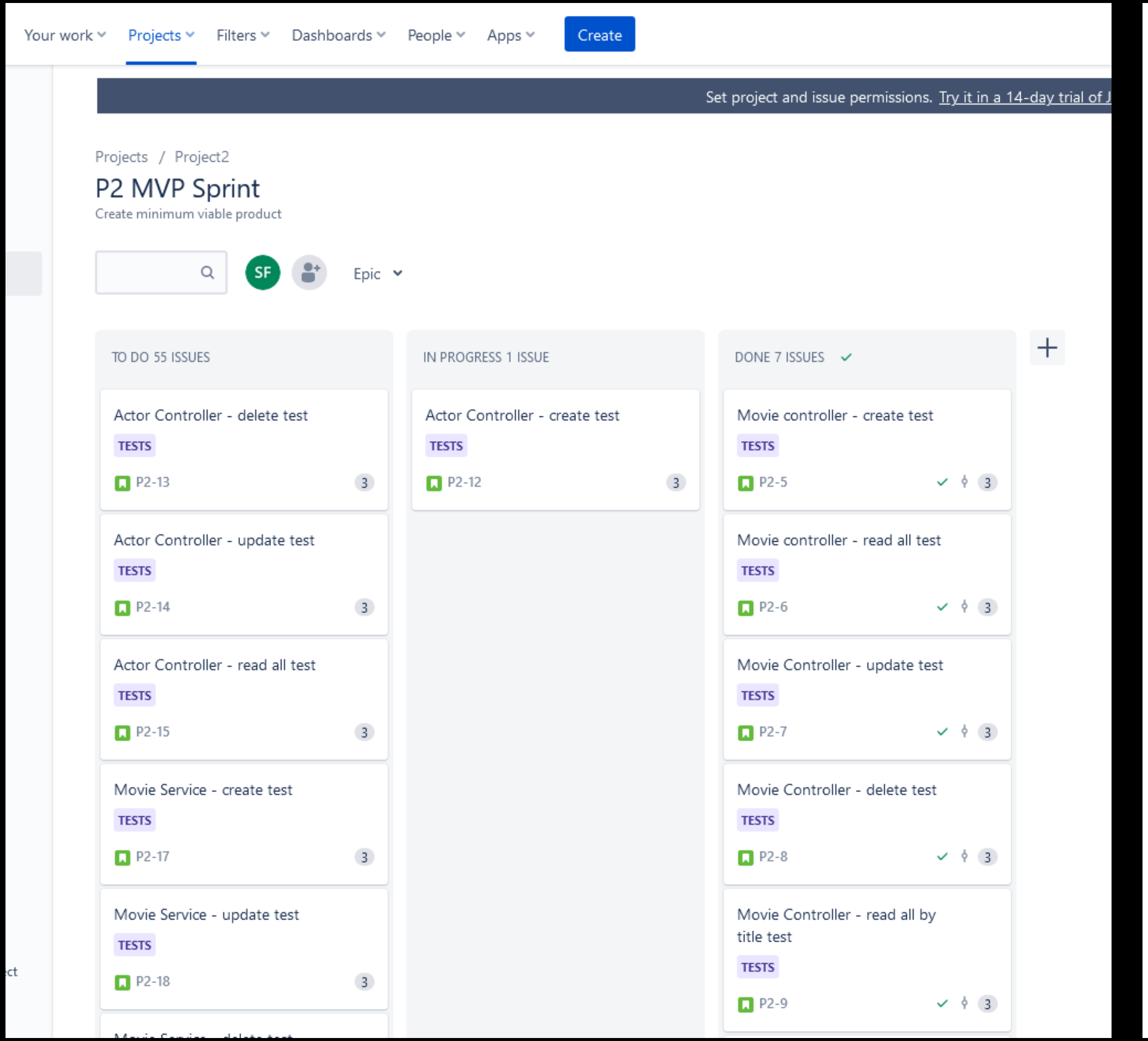
JIRA

Planning

- Prioritisation
  - Development separated in to sprints according to MoSCoW prioritisation
    - Initial MVP sprint
    - Additional Sprint for additional features
      - Image uploading
    - Features that were unable to be implemented
      - Sorting
      - Favourites list
    - Features that were not implemented dues to low prioritisation
      - User accounts

JIRA

Planning



Set project and issue permissions. Try it in a 14-day trial of J

Projects / Project2

P2 MVP Sprint

Create minimum viable product

TO DO 55 ISSUES

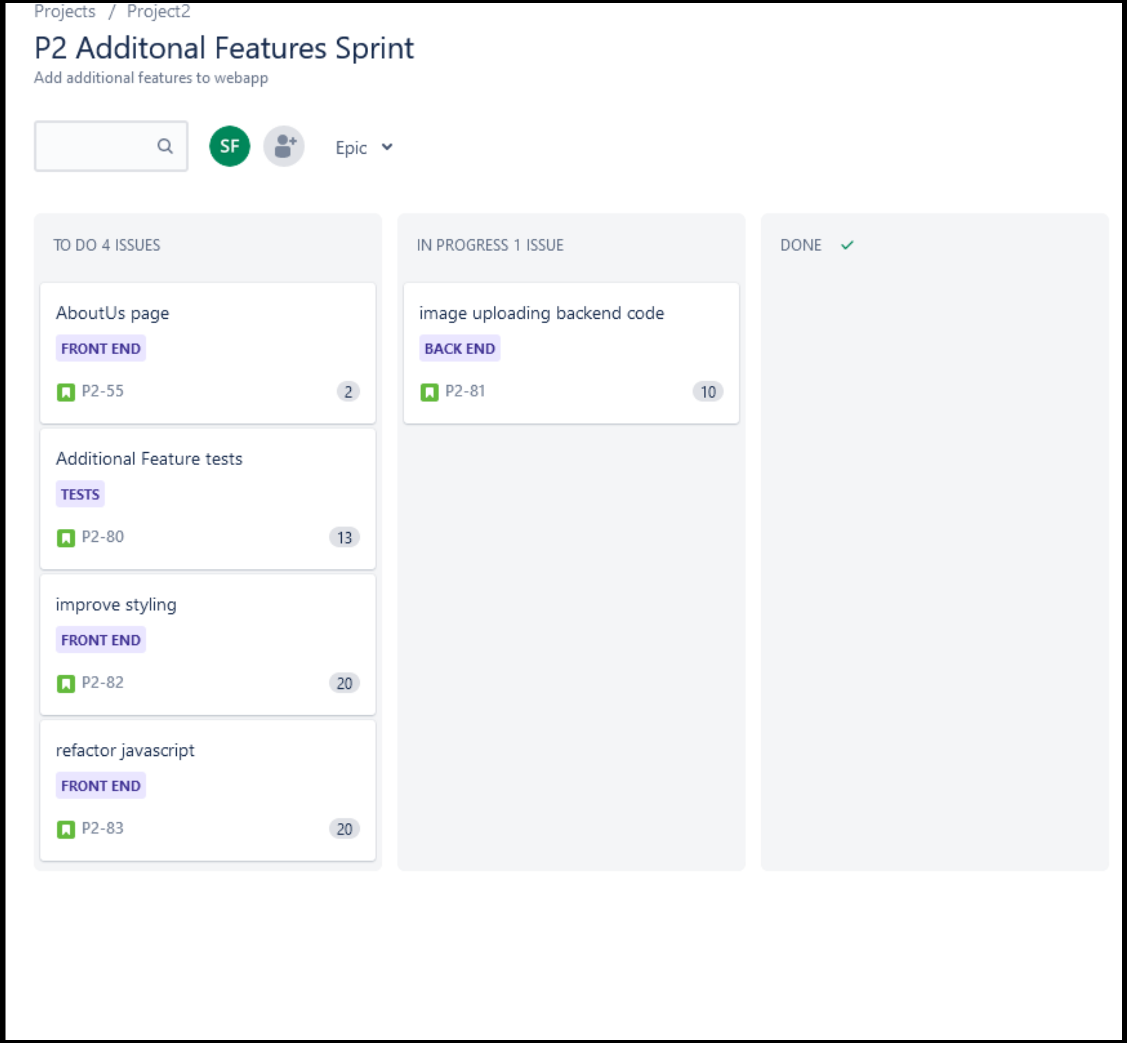
- Actor Controller - delete test, TESTS, P2-13, 3
- Actor Controller - update test, TESTS, P2-14, 3
- Actor Controller - read all test, TESTS, P2-15, 3
- Movie Service - create test, TESTS, P2-17, 3
- Movie Service - update test, TESTS, P2-18, 3
- Movie Service - delete test

IN PROGRESS 1 ISSUE

- Actor Controller - create test, TESTS, P2-12, 3

DONE 7 ISSUES ✓

- Movie controller - create test, TESTS, P2-5 ✓ 3
- Movie controller - read all test, TESTS, P2-6 ✓ 3
- Movie Controller - update test, TESTS, P2-7 ✓ 3
- Movie Controller - delete test, TESTS, P2-8 ✓ 3
- Movie Controller - read all by title test, TESTS, P2-9 ✓ 3



Projects / Project2

P2 Additonal Features Sprint

Add additional features to webapp

TO DO 4 ISSUES

- AboutUs page, FRONT END, P2-55, 2
- Additional Feature tests, TESTS, P2-80, 13
- improve styling, FRONT END, P2-82, 20
- refactor javascript, FRONT END, P2-83, 20

IN PROGRESS 1 ISSUE

- image uploading backend code, BACK END, P2-81, 10

DONE ✓

JOURNEY

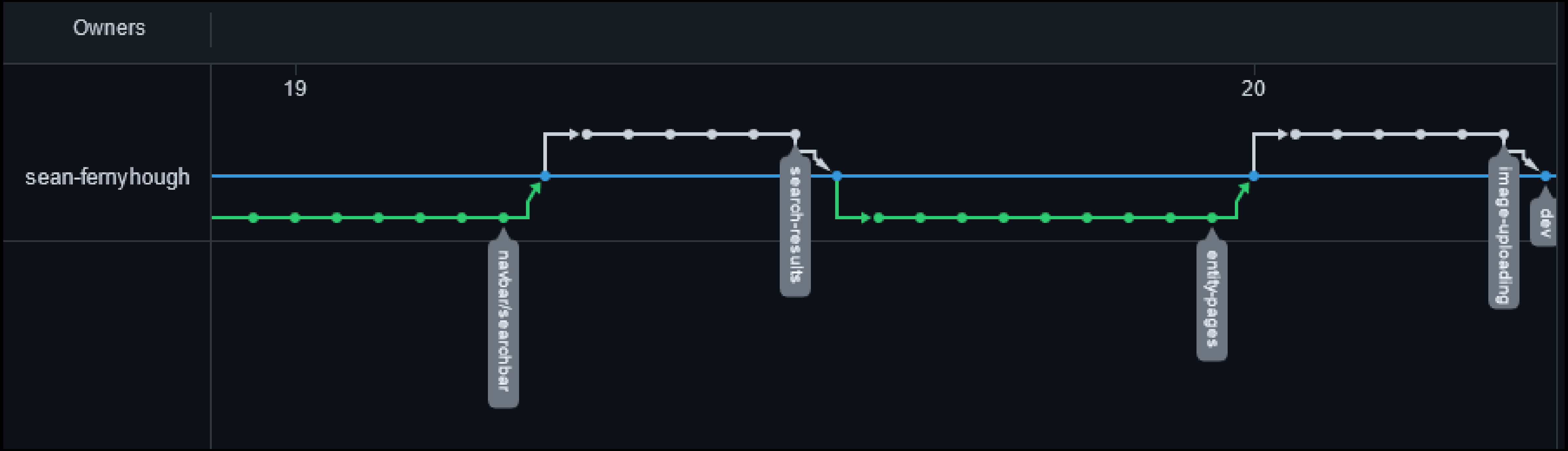
- Maven – build tool
- Spring/Spring Boot – backend framework
- Junit, Mockito – automated unit testing
- Selenium – automated frontend testing
- Java – backend
- HTML/CSS/Javascript - frontend

- JIRA – project management

- Git – version control

CONTINUOUS INTEGRATION

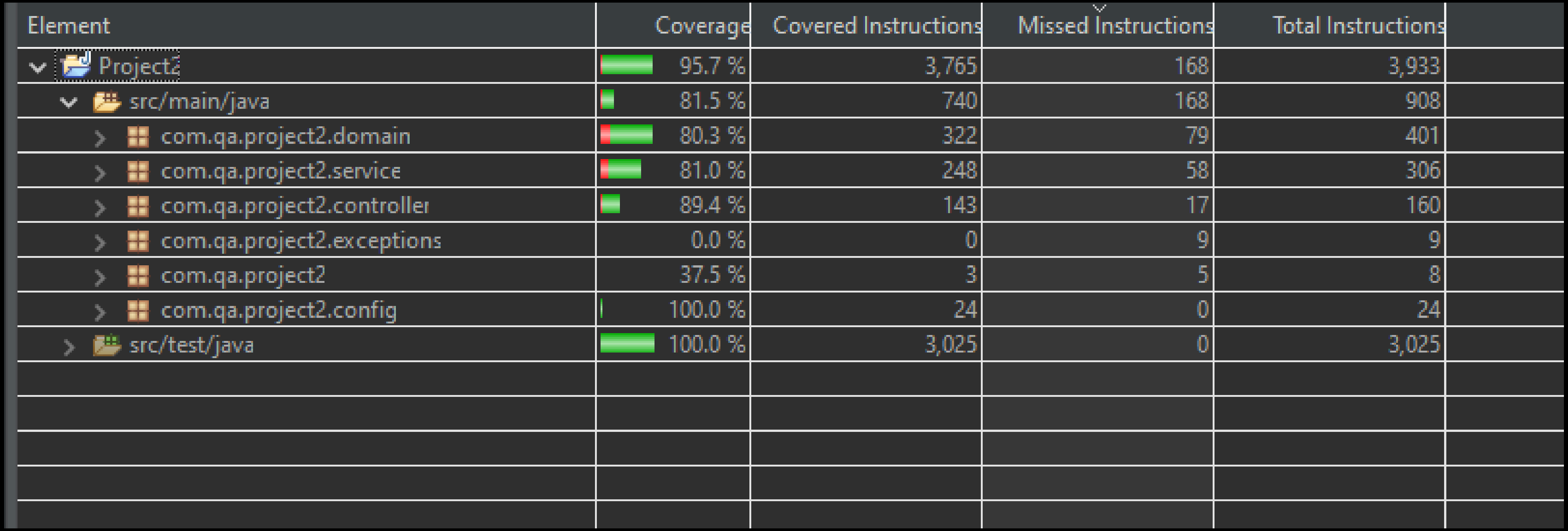
- Git for version control
  - Feature-branch model



TESTING

- Use of Junit, Mockito, selenium (51 tests in total, 8 of which are selenium tests)

- Total src/main/java test coverage of 81.5%
  - Tested all methods within domain, controller, and DAO for customer, item, and order, small amount of testing on IMS



Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Project2	95.7 %	3,765	168	3,933
src/main/java	81.5 %	740	168	908
com.qa.project2.domain	80.3 %	322	79	401
com.qa.project2.service	81.0 %	248	58	306
com.qa.project2.controller	89.4 %	143	17	160
com.qa.project2.exceptions	0.0 %	0	9	9
com.qa.project2	37.5 %	3	5	8
com.qa.project2.config	100.0 %	24	0	24
src/test/java	100.0 %	3,025	0	3,025

DEMONSTRATION

- A short Demonstration
- User Stories
  - A user wants to find an actor by name
  - A user wants to create a new movie
  - A User wants to remove an actor from a film and delete them

SPRINT RETROSPECTIVE

What went well?

- Test-driven development
- Prioritisation

What could be improved?

- Focus work on single files

SPRINT REVIEW

- M – Crud functionality was added for movies and actors, advanced search options, 80% test coverage minimum
- S – Added ability to add images to movies for improved user experience
- C – Would of liked to add sort options to make searching through large databases easier
- W – Chose not to add user account entities and front end functionality due to time constraints

CONCLUSION

- Learned from previous project
  - More effective planning – better use of prioritisation
  - Test-driven development – lead to producing working guide faster and gave more direction

- Project specific
  - Learned a wide variety of techniques
    - spring resource handling, sending multipart file form data
    - More effective asynchronous programming
    - Practiced sophisticated testing

Q&A

Thank you for listening.

Any Questions?