CRUD-based web application

LIBRARY

BY

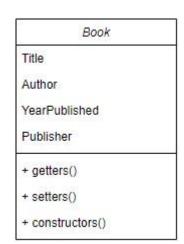
JORDAN

BICKERDYKE

Intro

- Approach
- Sprint plan
- Technologies used
- GitHub
- Testing
- Demo
- Sprint review and retrospective
- Close



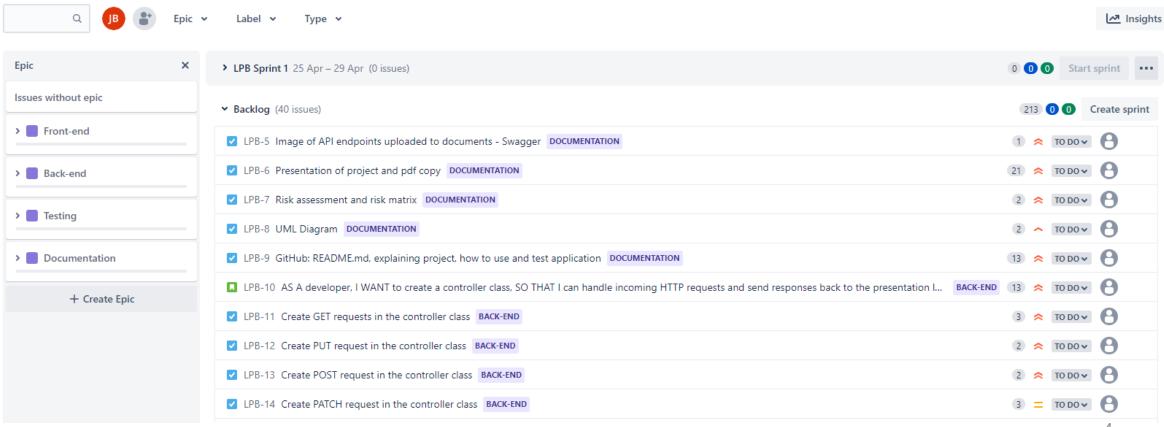


- Initial idea
- Plan the project using Jira as the project management board:
 - Create epics, user stories and tasks
 - Estimations using story points
 - Prioritise using the MoSCoW methodology
- Achieve MVP quickly
- Testing to >70%
- Custom queries, exception handling and better design if possible

Backlog

Projects / Library Project - BAE14

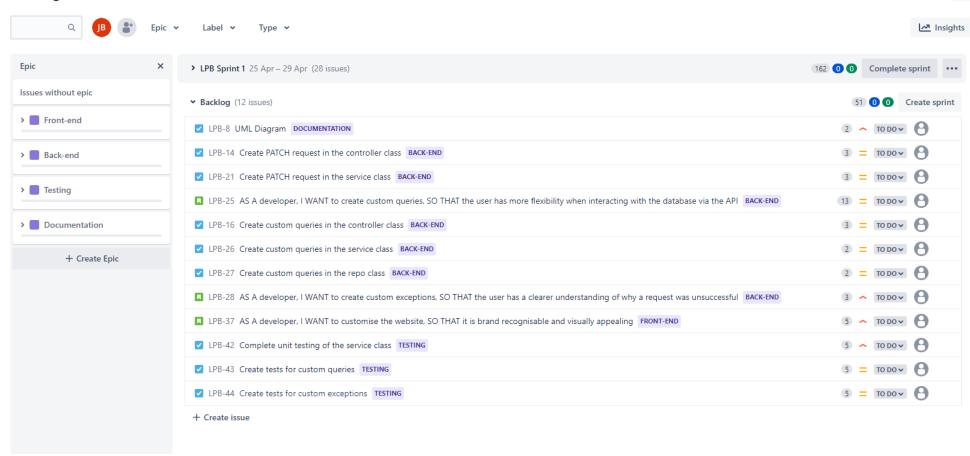
Backlog



Backlog

Projects / Library Project - BAE14

Backlog



Risk

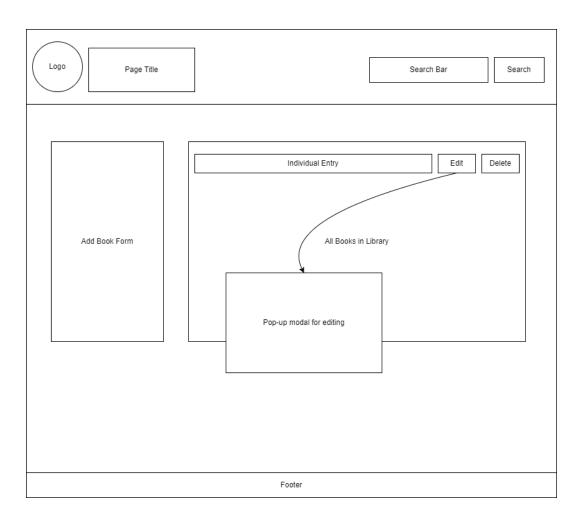
Description	Evaluation	Likelihood	Impact Level	Responsibility	Response	Control Measures
Developer becomes unwell/ill	Project would be incomplete	Low	High	Developer/Divine Entity	Ask for an extension	Hand hygeine and avoid extreme sports
Failure to meet the MVP	Fail the project	Low	High	Developer	Weep	Plan the project thoroughly with prioritoisation of tasks and asking for help from trainers where required
Hardware failure	Lost work and inability to complete project	Low	High	Developer	Inform trainers of issues, source hardware from elsewhere	Ensure work regularly backed up to remote repo
Cannot find solution to problem	Time lost	High	Low	Developer	Help from trainers	Look through course content and notes, search for solutions online on StackOverflow
Unable to achieve sufficient test coverage	Lost marks	Medium	Low	Developer	He <mark>l</mark> p from trainers	Attend workshop on testing, research testing and look through course content
Tests fail	Methods/software not working as expected	Medium	Medium	Developer	Evaluate tests and methods to determine where it is failing and correct as required	Write method and test code carefully. Could implement test-driven development or test methods as created
Methods don't work as expected	Project would not function as required	Medium	High	Developer	Correct method and code	Test methods, check methods working as expected as methods created
Incorrect user input	API stop working due to exceptions	High	High	Developer	Refactor methods to throw exceptions	Develop code to expect exceptions as the project is built

Impact	High	Developer becomes unwell/ill Failure to meet the MVP Hardware failure	Methods don't work as expected	Incorrect user input		
	Medium		Tests fail			
	Low		Unable to achieve sufficient test coverage	Cannot find solution to problem		
		Low	Medium	High		
		Likelihood				

Table Design

```
Book.java X J BookRepo.java J BookService.java
10 import javax.validation.constraints.Max;
11 import javax.validation.constraints.Min;
12
13 @Entity
14 public class Book {
15
16
17
189
      @Id
      @GeneratedValue(strategy = GenerationType.IDENTITY)
19
20
      private long id;
21
      @Column(nullable = false)
220
      private String title;
23
24
250
      @Column(nullable = false)
      private String author;
26
280
       @Column
29
      private int publishedYear;
30
310
       @Column
      private String publisher;
32
      @Column(nullable = false)
340
       @Min(1)
       @Max(5)
      private int rating;
38
       // Constructors ------
40
410
      public Book() {
          super();
```

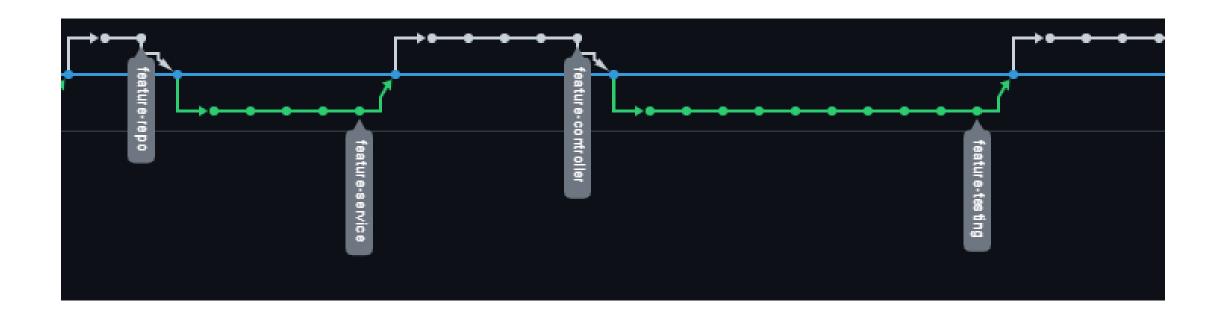
Frontend Design



Technologies Used

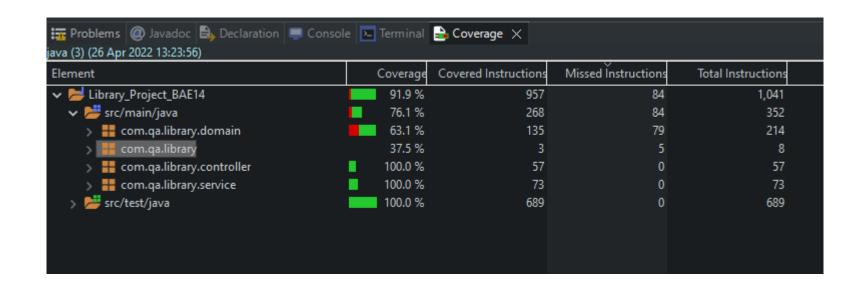
- Version Control System = Git
- Source Control Management = GitHub
- Project Management Board = Jira
- Database Management System = MySQL Server 8.0
- Back-End Programming Language = Java
- API Development Platform = Spring
- Front-End Technologies = HTML, CSS, JavaScript
- Build Tool = Mayen
- Testing = Junit, Mockito

Version Control



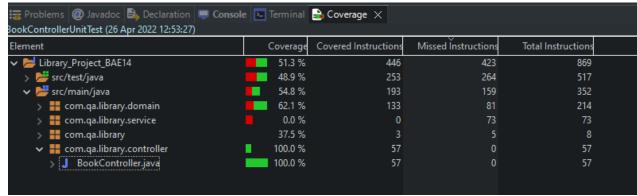
Testing

Coverage

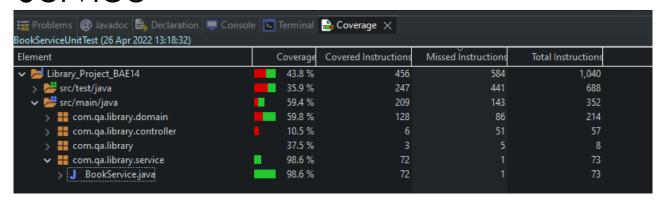


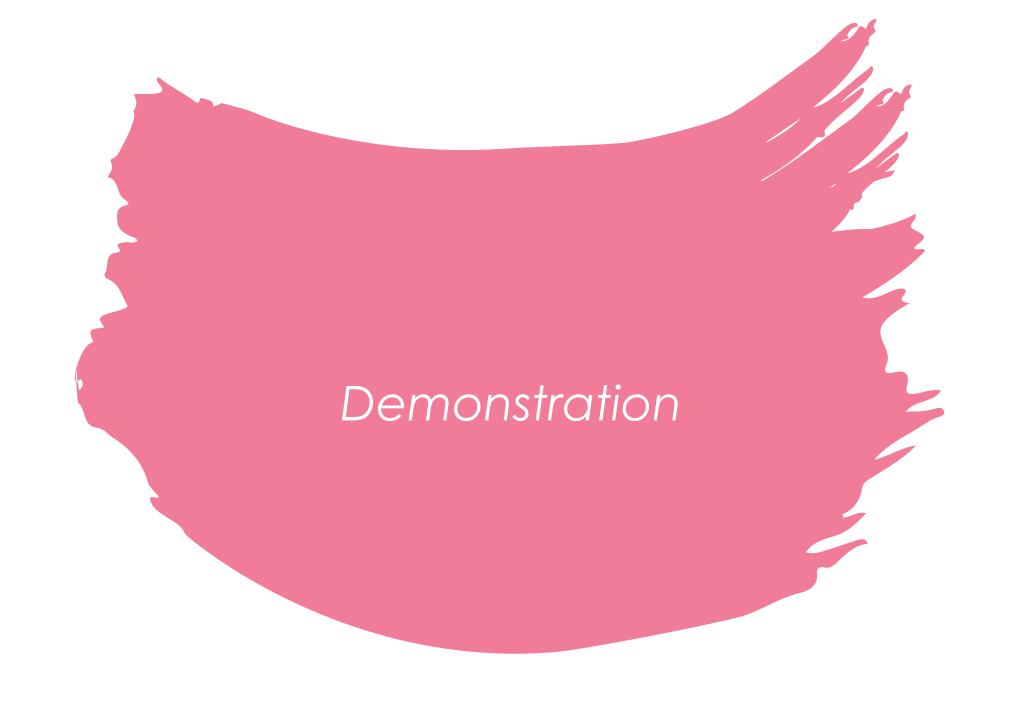
Unit Testing

Controller



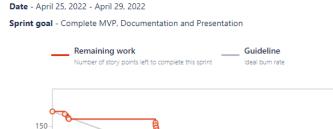
• Service





Sprint Review

- Sprint 1:
 - 28/28 issues completed
 - 162 story points
- Burndown:



12:00 26 Apr

12:00 27 Apr

12:00 27 Apr

12:00 28 Apr

12:00 28 Apr

12:00 29 Apr

12:00 29 Apr

120

30

12:00 26 Apr

Sprint Retrospective

What went well?

• What went 'less' well?

Thank you

Questions?