# CRUD-based web application

LIBRARY

BY

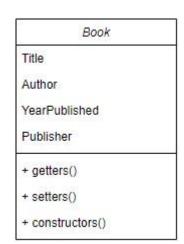
**JORDAN** 

**BICKERDYKE** 

#### Intro

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



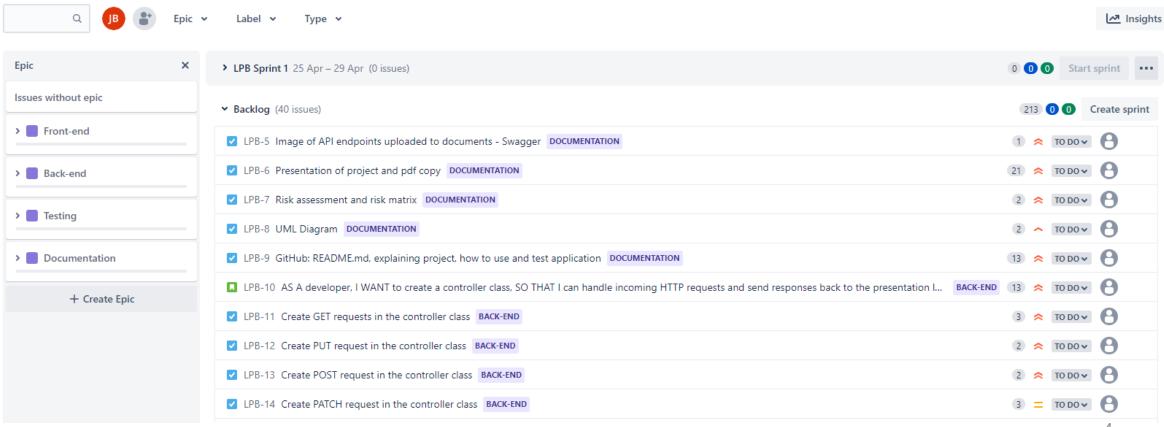


- Initial idea
- Plan the project using Jira as the project management board:
  - Create epics, user stories and tasks
  - Estimations using story points
  - Prioritise using the MoSCoW methodology
- Achieve MVP quickly
- Testing to >70%
- Custom queries, exception handling and better design if possible

## Backlog

Projects / Library Project - BAE14

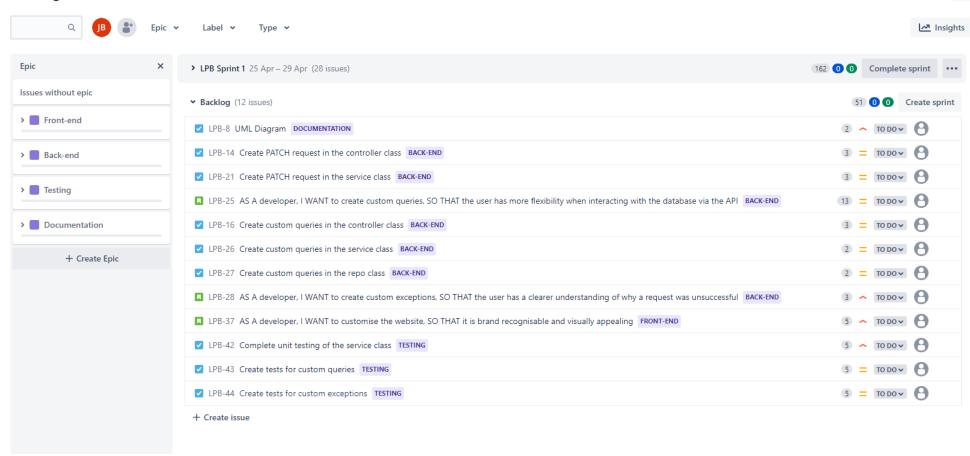
#### Backlog



## Backlog

Projects / Library Project - BAE14

#### Backlog



## Risk

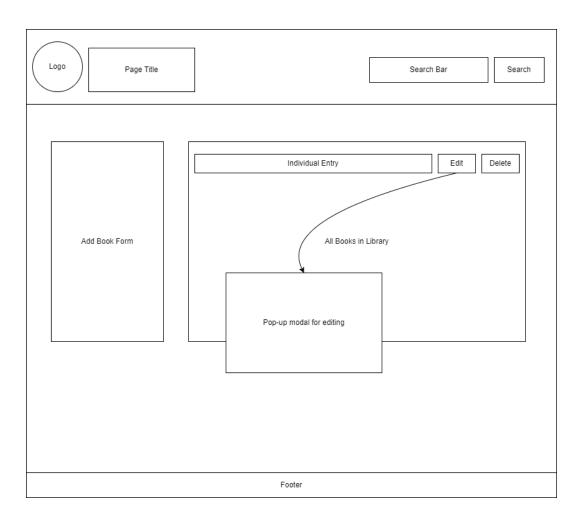
Description	Evaluation	Likelihood	Impact Level	Responsibility	Response	Control Measures	
Developer becomes unwell/ill	Project would be incomplete	Low	High	Developer/Divine Entity	Ask for an extension	Hand hygeine and avoid extreme sports	
Failure to meet the MVP	Fail the project	Low	High	Developer	Weep	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 wo expected as methods create		
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	

	High	Developer becomes unwell/ill Failure to meet the MVP Hardware failure	Methods don't work as expected	Incorrect user input			
Impact	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();
```

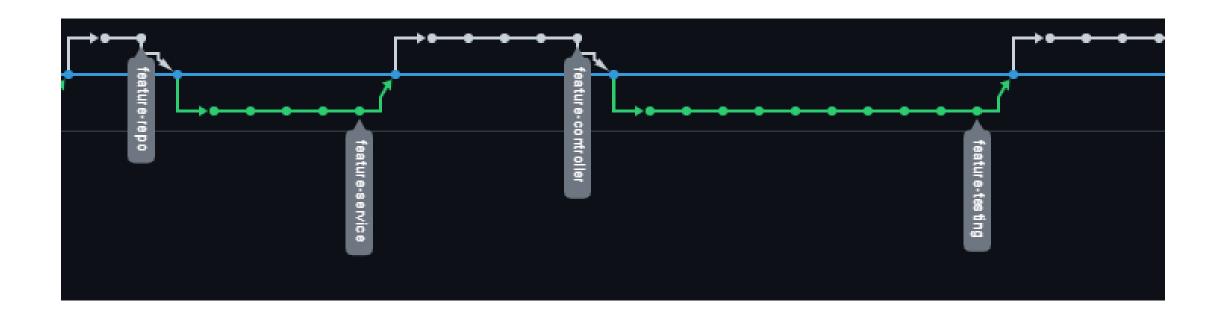
## Frontend Design



#### Technologies Used

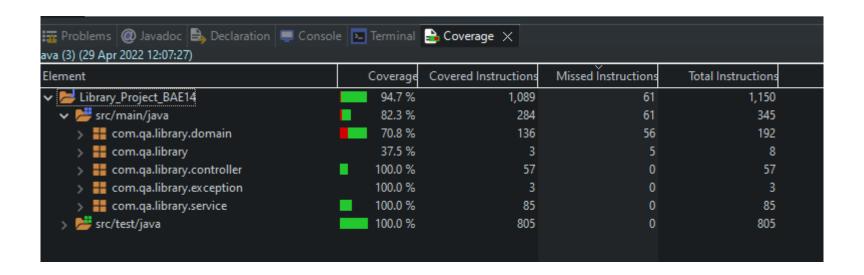
- Version Control System = Git
- Source Control Management = GitHub
- Project Management Board = Jira
- Database Management System = MySQL Server 8.0
- Back-End Programming Language = Java
- API Development Platform = Spring
- Front-End Technologies = HTML, CSS, JavaScript
- Build Tool = Mayen
- Testing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

#### Version Control



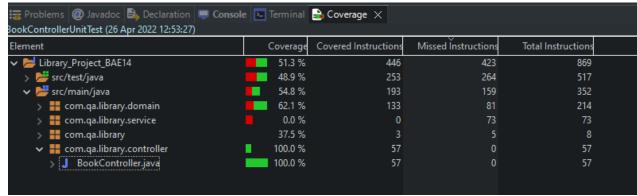
## Testing

#### Coverage

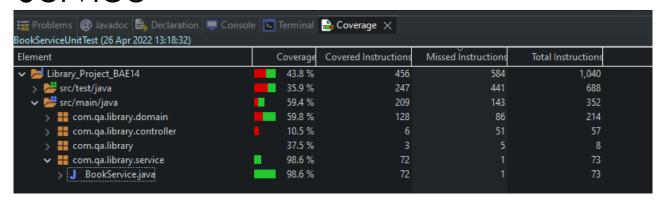


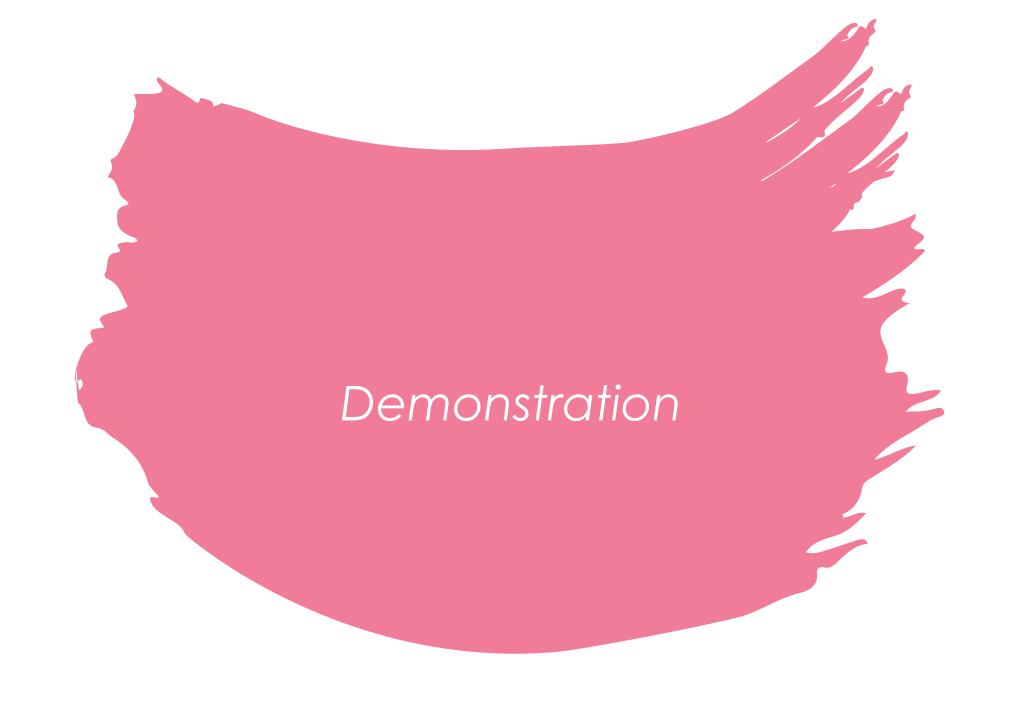
#### Unit Testing

#### Controller



#### • Service





#### Sprint Review

- Sprint 1:
  - 28/28 issues completed
  - 162 story points
  - MVP only (MUST have)
- Burndown:



Date - April 25, 2022 - April 29, 2022

### Sprint Review

- Sprint 2:
  - 8/8 issues completed
  - 38 story points
  - SHOULD/COULD have

Completed issues View in issue navigator

Key :	Summary÷	Issue type:	Epic :	Status :	Assignee	Story points
LPB-25	AS A developer, I WANT to create custom queries, SO	Story	BACK-END	DONE		13
LPB-16	Create custom queries in the controller class	Task	BACK-END	DONE		3
LPB-26	Create custom queries in the service class	✓ Task	BACK-END	DONE		2
LPB-27	Create custom queries in the repo class	✓ Task	BACK-END	DONE		2
LPB-28	AS A developer, I WANT to create custom exceptions,	Story	BACK-END	DONE		3
LPB-42	Complete unit testing of the service class	✓ Task	TESTING	DONE		5
LPB-43	Create tests for custom queries	✓ Task	TESTING	DONE		5
LPB-44	Create tests for custom exceptions	✓ Task	TESTING	DONE		5

Sprint Estimation field

LPB Sprint 2 Story points 

Story points

Date - April 29, 2022 - April 29, 2022



### Sprint Retrospective

What went well?

• What went 'less' well?

### Thank you

Questions?