

BOOKEROO

SEPT - GROUP 02

Members: Justin John, Daniel Marmion, Rylan Koneri, Mohammed Mousa

Contents

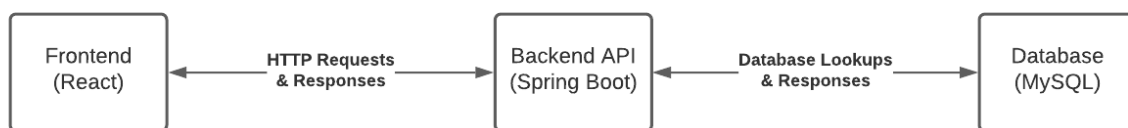
Vision Statement	2
System Architecture	2
Refactoring	2
Gitflow Organisation	3
Scrum Process	3
Deployment Pipeline	4
Testing	4

Vision Statement

Our vision for this product is to provide an easily accessible bookstore for bookworms around the globe. For those that enjoy reading in their free time who need a variety of books to read from, Bookeroo is an online book sharing/selling application that allows users and businesses to sell/buy books on the app highlighting a big catalogue of books available on the application for users to purchase from.

Our product is valuable even though it's minimalistic, with its easy to access user interface minimizing any clutter on each webpage and only showing what is needed for the user to choose from on the designated page they are on. This helps ensure that readers can effortlessly view and purchase books without any interference or hassle affecting their experience on Bookeroo. What makes Bookeroo more valuable than other online bookstores is its wide, dynamic catalogue of books to choose from which can easily increase in size as more books get posted up on the platform. This safeguards the motto that our product fulfills, being that 'there really is something for everyone on Bookeroo' showcasing different genres and prices of the books that are available for all users.

System Architecture/Design



Refactoring

The search bar initially looped through twice in retrieving the collection of books and for getting count, this was necessary due to the flow of code and due to how react hooks worked. A solution to remove this duplication was found and implemented.

Another major refracting done was the change from using the H2 database to MySQL. This implementation was moved from the submission for milestone 2 to milestone 3 due to the change in requirements of the assignments. The team decided it was best to focus on requirements that were more important for milestone 2 submission since it had higher priority.

Console logging is another refactoring undertaken for the final submission. Logs have been placed in important aspects of the code to see output where necessary.

General clean-up of the code was also undertaken throughout the project, to remove duplication, add comments for clarity or just to make the project better organised.

Gitflow Organisation

We managed our Git repository by following the standard Gitflow organisation. We had a master branch reserved for releases, and a develop branch that maintained the current version of the software, with features added to this branch as they were finished. Whenever we created a feature, we created a new branch, created the feature on that branch, then made a pull request to merge that branch back into develop when that feature had been completed. After the code had been reviewed, it would be merged into develop and that extra branch would be deleted. A release was made to the master branch once per milestone, at the end of each milestone.

We committed our code regularly as we worked on it, with there often being several or many commits on at least one branch each day.

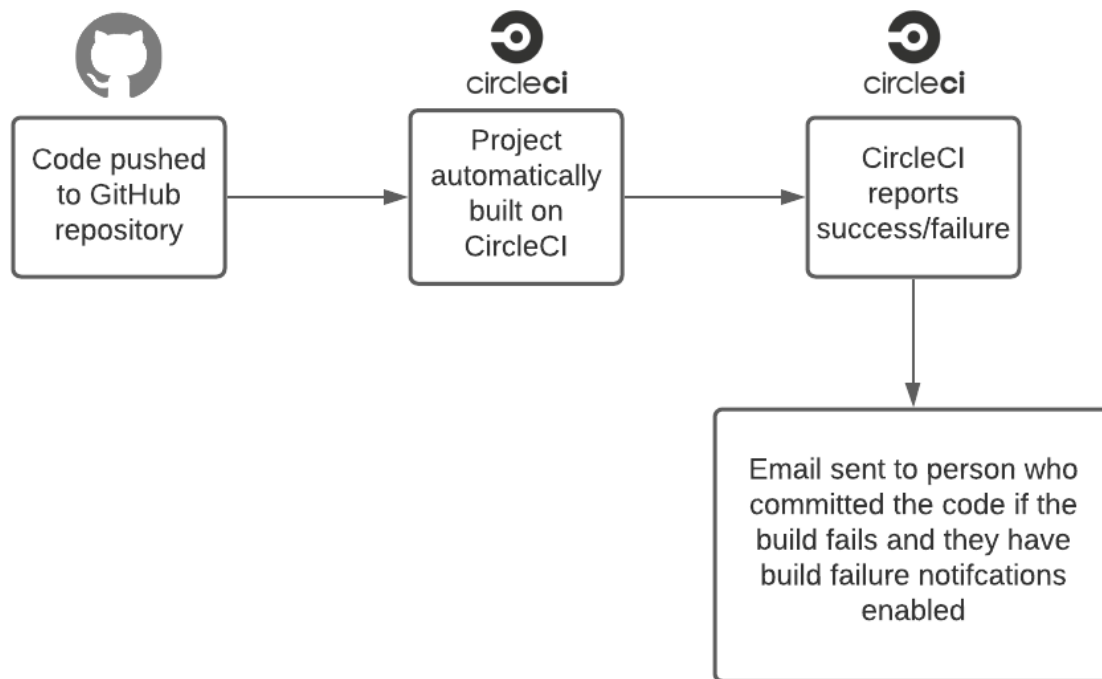
Scrum Process

Our Scrum Team was made up of 4 members, Mohammed being the Scrum Master and Daniel, Justin and Rylan being members of the Scrum Team along with our tutor Dipto being the Product Owner.

Our weekly quota of meetings were two a week which was fulfilled most of the time, where the first meeting was usually on the weekdays and the second meeting on the weekend. Everyone was present for each meeting as the meeting times were flexible ensuring no one in the Scrum Team misses out on any discussion of the product.

There were 5 sprints, the first being essentially an intro sprint going over roles and planning of implementing the product, sprints 1, 2 and 3 mainly implementing the features discussed in the user stories chosen and the final sprint going over refactoring of any code and finalizing our final product along with ensuring all documentation is present. At the start of each sprint the first meeting was essentially the planning meeting, preparing the sprint backlog and assigning tasks and at the end of each sprint the last meeting is the retrospective meeting going over the retro notes for the sprint, discussion of how the efficiency of the sprint was in terms of work completed and how we can improve next sprint. Our platform of choice to have meetings was on Discord as this worked for all of the Scrum Team and it makes it easy to join a voice call from any device.

Deployment Pipeline



Testing

Acceptance Testing:

Within the acceptance testing procedure regarding Milestone 3, we were able to perform tests on 8 different user stories. Performing these acceptance tests allowed us to determine whether the software system has met the requirement specifications. The main objective of these tests were to evaluate the system's capability and compliance with the requirements and verify if it has met the required criteria for delivery to end users. Each of the tests focused on the functionality provided and aimed to simulate potential user experiences based on the user stories provided. The tests ensured that all functionality was tested and that there were no outstanding bugs on the software which met requirements.

Majority of tests were completed, resulted in full completion and there were no bugs found in those processes. Users are able to utilise the search/navigation bar effectively and are able to enter details such as (book title, author name, ISBN), which gives them their desired book choice to select. Furthermore, users are able to visually view the book through an “individual book” page, which further gives users ability to dive into the table of contents of the book according to their liking.

However, there were bugs found regarding a user's attempt to login and for their credentials to be verified. Users are able to enter their details through a given textbox, however their credentials are unable to be verified via the database, clicking the submit button will automatically take the user to the homepage.

Conclusively, it can be denoted that the majority of the test passed the requirements and criteria for the software, however some bugs were located within the process of the user logging in. The feedback received from these acceptance tests allow us to fixate on the things that were completed effectively and additionally focus on further desired improvements, before the system is fully released.

Evidence of Testing

The screenshot displays a REST client interface with a GET request to `localhost:8080/api/v1/books/all`. The response status is 200 OK, with a time of 510 ms and a size of 15.03 KB. The response body is shown in JSON format, containing a single book object.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
{
  "id": 1,
  "bookName": "A Tale of Two Cities",
  "author": "Charles Dickens",
  "description": "The novel tells the story of the French Doctor Manette, \n          his 18-year-long imprisonment in the Bastille in \n          Paris \n          and his release to live in London with his daughter Lucie,\n          whom he had never met. The story is set \n          against the conditions \n          that led up to the French Revolution and the Reign of Terror.",
  "cost": 25.0.
}
```

(continues over next pages)

localhost:8080/api/v1/users/register

POST localhost:8080/api/v1/users/register

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "username": "admin@resonatingreads.com.au",
3   "accountType": "pendingpublisher",
4   "fullName": "Resonating Reads",
5   "address": "256 Collins Street, Melbourne",
6   "phoneNumber": "8395228502",
7   "password": "bookworm",
8   "confirmPassword": "bookworm",
9   "abn": "95827121617"
10 }
```

body Cookies Headers (14) Test Results

Status: 201 Created Time: 823 ms Size: 802 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "username": "admin@resonatingreads.com.au",
4   "accountType": "pendingpublisher",
5   "fullName": "Resonating Reads",
6   "address": "256 Collins Street, Melbourne",
7   "phoneNumber": "8395228502",
8   "password": "$2a$10$FF5mdkHgt9SnApLZPw1074j./Cuy158fyF3PmWpJ3aAudAa/xQY.",
9   "confirmPassword": ""
10 }
```

GET localhost:8080/api/v1/books/a

Params Authorization Headers (8) Body Pre-request Script

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2021-10-23T09:02:39.410+00:00",
3   "status": 404,
4   "error": "Not Found",
5   "message": "",
6   "path": "/api/v1/books/a"
7 }
```

```
[INFO] Results:
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ loginmicroservices ---
[INFO] Building jar: C:\Users\dmarm\Documents\Code\Uni - Software Engineering Process and Tools\Bo
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.2.RELEASE:repackage (repackage) @ loginmicroservices ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 11.838 s
[INFO] Finished at: 2021-10-23T20:01:29+11:00
[INFO]
```

```
PASS src/utls.test.js
PASS src/components/Layout/Footer.test.js
PASS src/components/Layout/Landing.test.js
```

```
Test Suites: 3 passed, 3 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        6.013s
Ran all test suites.
```

```
Watch Usage: Press w to show more.█
```