Project Report

SEPT – COSC2299 – Group 4 Milestone 3

# Team Info

Team Name: Group 4

## Links

GitHub: <https://github.com/Someoneamzing/sept-group-assignment>

JIRA: <https://septassignmentproject.atlassian.net/jira/software/projects/SEP>

CircleCI: <https://app.circleci.com/pipelines/github/Someoneamzing/sept-group-assignment>

AWS Deployment: <http://cicddevelopment.s3-website-ap-southeast-2.amazonaws.com/>

## Team Members

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Task List |
| Minggang Dong | 25 % | Worked on sep-91  -add a filter page with a list of button for the genres  -add sql query for getting the filtered books, and controller to receive the genre  -let the front-end display only the selected genres books  -add both back-end and front-end tests.  Refactored the book api to use the axios instance. |
| Moritz Hauptmann | 25 % | Worked on SEP-17, SEP-18,  SEP-36  -Added admin user management functionality, seeing all users in a list, accepting business user requests, and editing user details, including suspending users and accepting them again  -Added users having their own profile with their user details  -Added functionality for user to edit their profile, including changing their password. |
| Jack Hollis-London | 25 % | Cried |
| Jacob McEwan | 25 % | Worked on SEP-7  -Created Order Microservice  -Added Order List frontend  -Added Order view frontend  -Added Order Items  -Wrote current tests for Order microservice and order frontend.  Refactored to include logging with Log4J |

## Scrum Organisation

**Scrum Master:** Jack Hollis-London

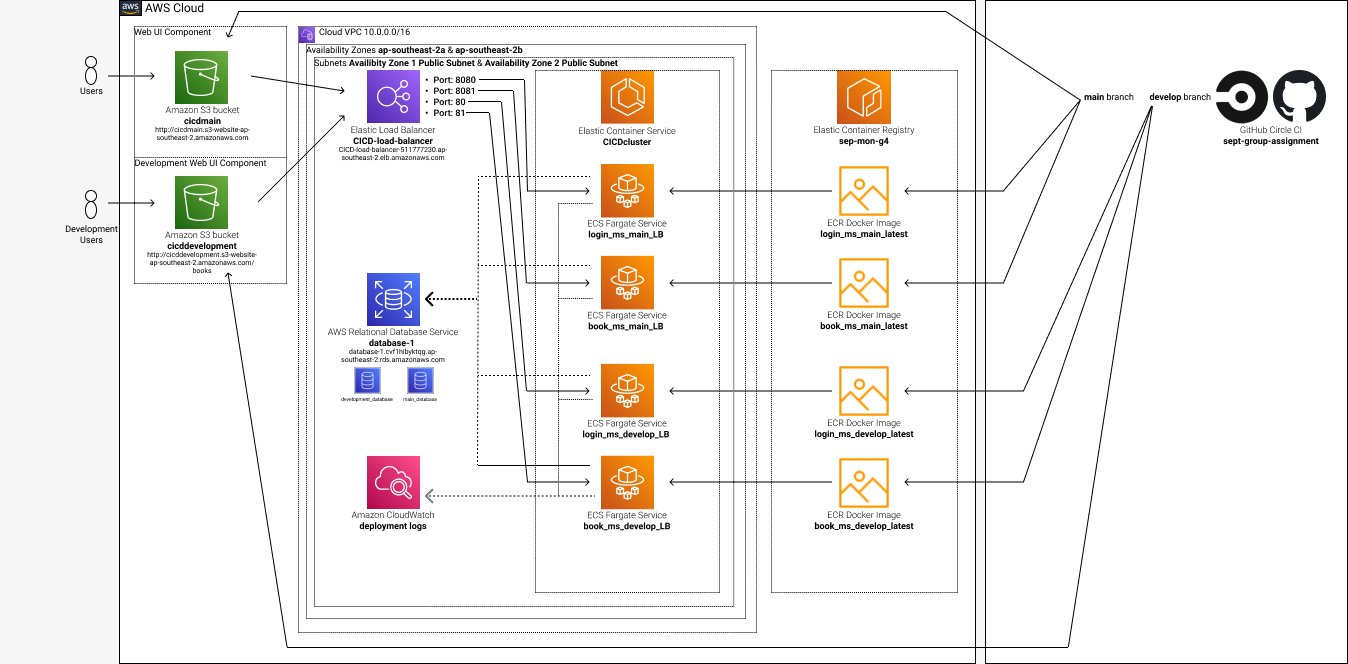
**Development Team:** Minggang Dong, Moritz Hauptmann, Jack Hollis-London, Jacob McEwan

Meetings were held regularly twice a week with extra meetings based on work progress and issues that popped up. A meeting was held every Monday and Tuesday due to availability of team members and individual meetings between 2- 3 members were held as needed for problem solving. Sometimes an extra meeting was held on a Friday or Saturday with the whole team to organise submissions or sprint planning / retro.

# Vision

Our vision was to create a book trading platform for anyone interested in books. We wanted people that share the same passion for books as us to have a place to find books that they want to read. Furthermore, with the rating and review system we tried to create a place for people to publicly share their opinion and help other users being able to decide if the book is the right one for them. If the user is still unsure whether the book is right one to get or not, users have the option to read the first few pages of each book, which is available in the preview of each book. With allowing businesses to register and sell new books we want to bring in new books to our platform. Registered businesses should be the only businesses to sell new books to ensure that only new books are being sold as new ones. A business is registering with an ABN to be liable for their transactions and for tax purposes. We are accepting business register requests by hand to have a trustworthy platform, with only from us as trustworthy classified business partners. Paying and ordering books should be as simple as possible for customers, therefore we are accepting PayPal payments and offering refunds within 2 hours of purchase. Moreover, transaction histories are available for customers as well as businesses in their dashboard. Our main goal is it to create a big platform for our users to find any book they are after, which is including some rare books that are only available as second-hand ones, because they are not being produced anymore.

# System architecture



# GitFlow

Throughout the project the team followed a GitFlow method for organising branches and commits. The ‘main’ branch was reserved for the latest release. Only full features ready for production are committed. ‘develop’ is where the current development version of the project resides. Every time new work is to be started, a new branch is made labelled with the type of work, e.g., feature, refactor or fix. The work is made in that branch and when finished a pull request is made. This pull request is then passed through the CI pipeline to ensure all tests pass. Then all members must review the pull request and approve it before it can be merged. Once approved it is merged into develop. When a release is made a new tag is made for the release and all the associated artefacts are attached.

# New Features

Below is a list of new user stories implemented and the features added in Milestone 3

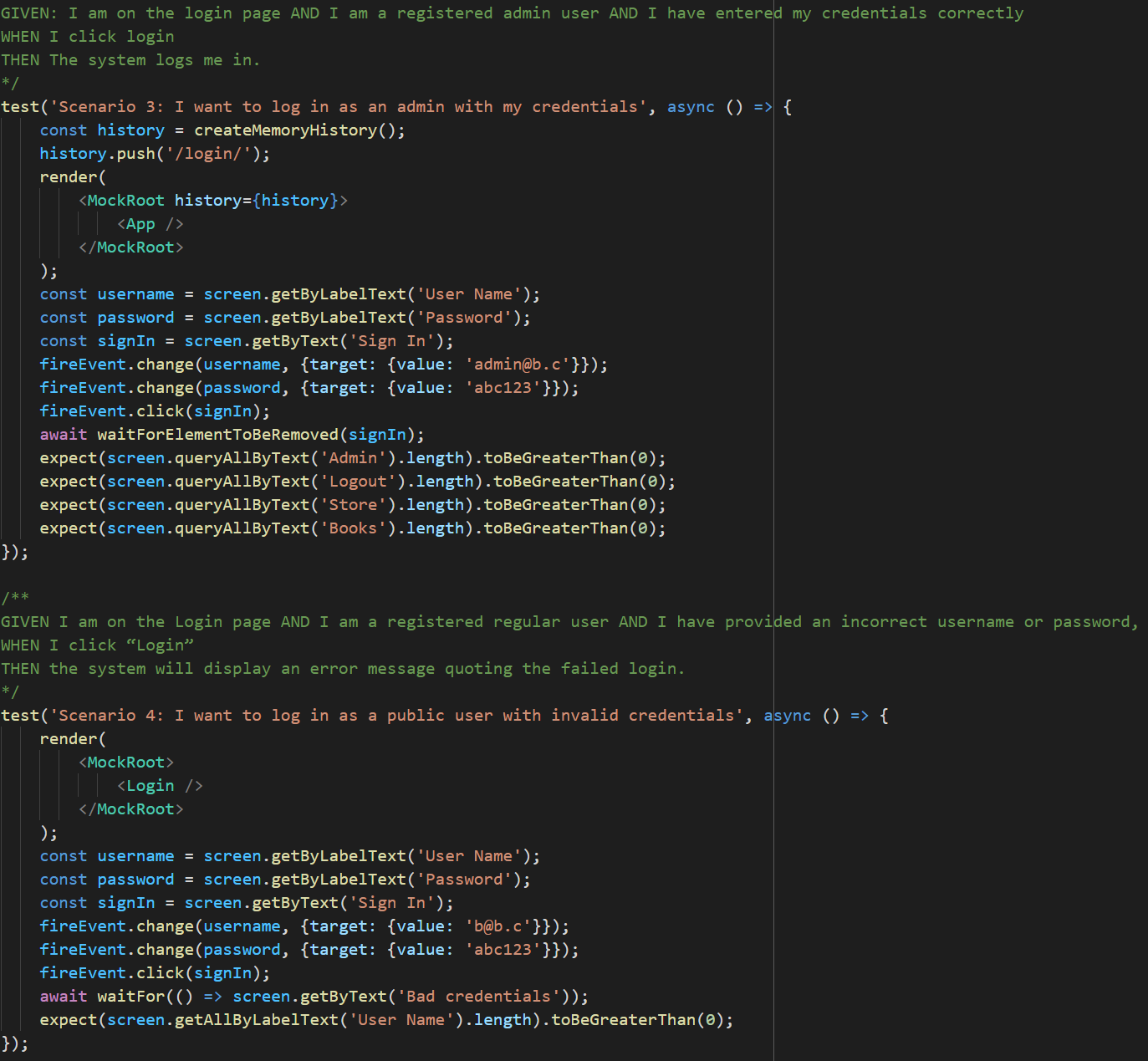
|  |  |  |  |
| --- | --- | --- | --- |
| PBI ID | User Story | Added Features | Added Tests |
| SEP-16 | As a business I want to register my business on bookeroo pending approval so that I can sell books on bookeroo | * Business User Registration | * sep-16: New User inputs empty ABN * sep-16: Successful registration request * sep-16: Navigate to the business register page from the public register. * emptyABN * successfulBusinessRegistration |
| SEP-91 | As a user, I want to filter out the books by their categories so I can see books with my favourite genre | * Books can be put in genres * Books can be viewed by genre | * getAllBooksOnFilterPage * getAllHorrorBooks * FilterBooksPage.test.js |
| SEP-17 | As an admin user, I want to approve/suspend users and pending business user registrations so that I can maintain the platform | * Approving / Rejecting Business User Applications * Suspending User Accounts | * accountIsLocked * businessAccountIsDisabled |
| SEP-18 | As a user, I want to edit users so I can moderate users on bookeroo | * Admins can remove inappropriate content in user profiles * Users can edit their profiles | * userProfileGet * userProfileUpdate * userProfileUpdateAuthenticationReject * businessUserUpdate * userUpdate |
| SEP-36 | As an admin user, I want to see all of bookeroo's users so I can manage them. | * Admins can see all the users on the platform |  |
| SEP-75 | As any type of user I want to see the list of books available so that I can choose what to buy |  | * ViewAllBooks.test.js |

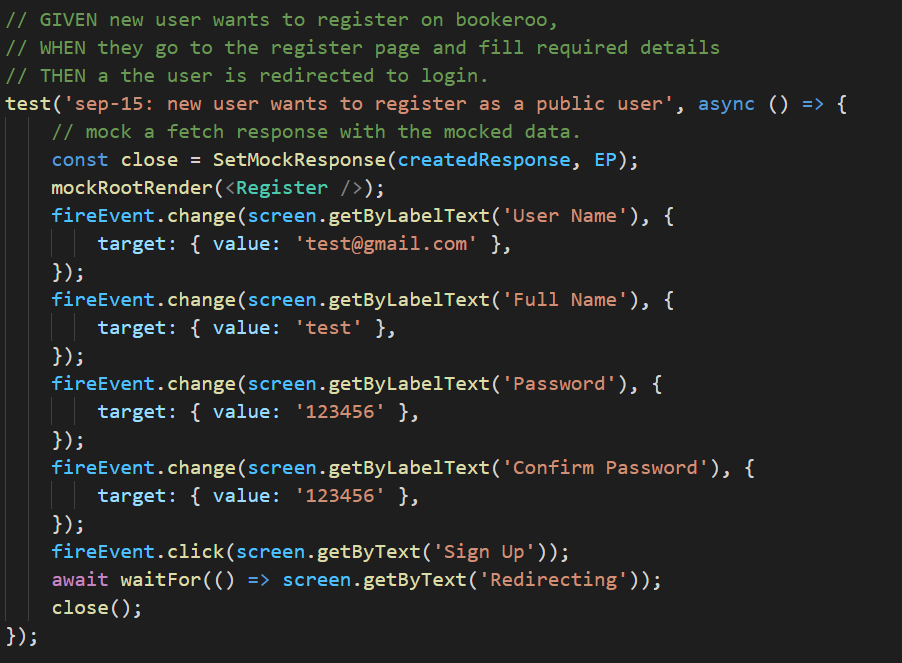
# Testing

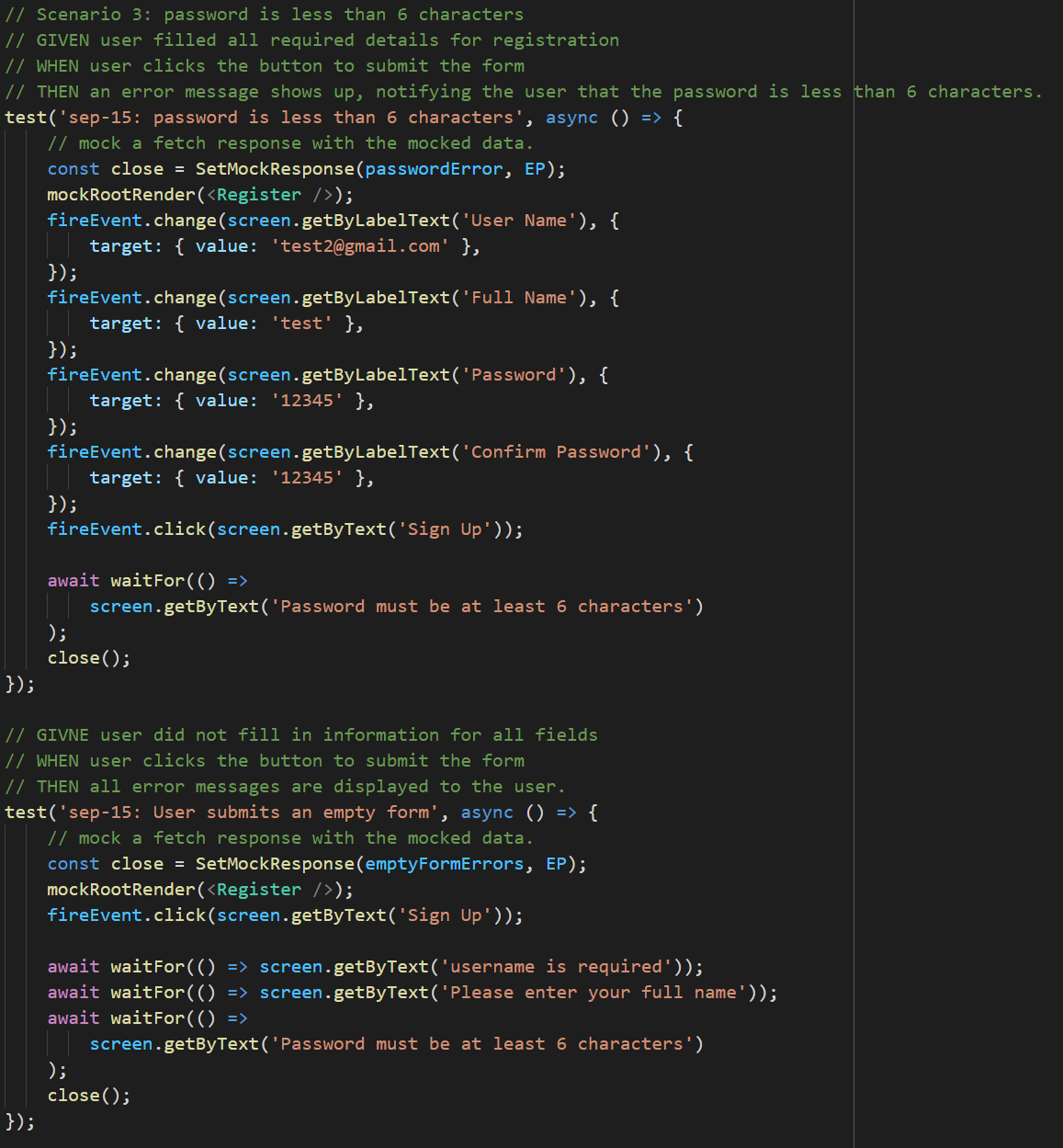
## Acceptance Testing

Below are some examples of the automated acceptance testing:





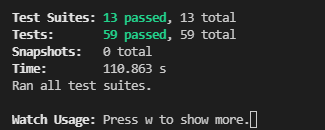




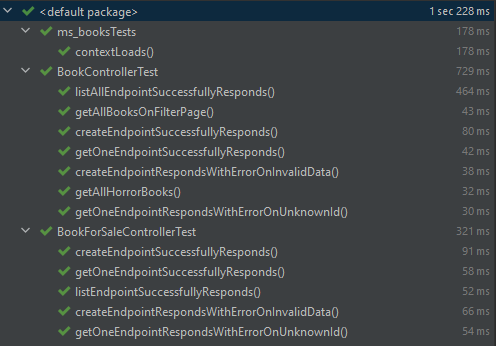
## Test Results

Below are screenshots of the test results of the frontend and backend code

Frontend:



Backend Book Microservice



Backend Login Microservice

