COSC 2299: SOFTWARE ENGINEERING PROCESS AND TOOLS

MAJOR PROJECT: MILESTONE 3

DOCUMENTATIONS

Team name: **SEPT Squad**

Tuan Vu (S3678491)

Phu Nguyen (S3598755)

Janidu Higgoda (S3846409)

Edward Kahiro Kuo (S3691466)

Production Application: http://sept-fe-alb-1073804615.ap-southeast-2.elb.amazonaws.com/login

Github link: https://github.com/s3516565/SEPT_TeamProject_Sem2_2021

Jira link: https://rmit-sem2-2020-sept.atlassian.net/jira/software/projects/SEPT/boards/1

Github Action: https://github.com/s3516565/SEPT_TeamProject_Sem2_2021/actions

Contents

SUMMARY OF NEW USE STORIES

- Total user stories completed in Sprint 3 and 4

- **US1**: As an Admin I want to be able to approve shop owners so that I can allow only legal users to be members.

- **US2**: As a public user/shop owner, I want to see my transaction history so that I can keep track of the books I have purchased or sold

- **US3**: As a public user I want to add a book to my cart and buy book online so that I don't have to go to the bookstore to buy books.

SUMMARY OF NEW FUNCTIONALITIES

Feature	Priority	Priority Order
Admin to approve shop owners	Critical	5
users/shopowners to access their transaction history	Critical	5
users to add books to the cart and purchase them online	Critical	8

PRODUCT BACKLOG

_

PRODUCT BACKLOG

Notes:
There are three types of user accounts: Admin, publisher/shop owner and public user.

ID	ITEM	OWNER	PRIORITY	EFFORT	STATUS	DEFINITION OF DONE
1	As a shop owner, I want to remove a book entry, so that I can have number of books under the limit.		High	3.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
2	As a Shop owner, I want to edit book's author information , so that the book's author information is correct.		High	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
3	As a Shop owner, I want to change the selling price of a book, so that I can sell the book to the customer at a new price.		High	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
4	As a shop owner, I want to change the status of a book to available, so that I can Sell the book to the customer.		High	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
5	As a shop owner, I want to change the status of a book, so that I can make the book unavailable to the customer.		High	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
6	**As an Admin, I want to search for transactions by month so that I can see how many books were sold on the platform each month.**		Medium	5.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
7	As an Admin, I want be able to approve or block new users so that I can allow only legal users to be members.		Critical	5.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
8	As an Admin I want be able to edit the details of books so that I can keep all the books up to date		Critical	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
9	As an Admin I want to be able to remove books that violate the rules so that the book can no longer be displayed.		High	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
10	As an Admin I want to be able to edit the details of a user so that the user's profile doesn't show any incorrect details		High	3.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
11	As an Admin I want To be able to add a new user so that Person can start buying or selling books.		Medium	3.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
12	As an Admin I want to be able to download report of a book So that I can inspect certain stats about that book		Medium	5.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
13	**As an Admin I want to be able to terminate accounts so that I can remove offending accounts. (after account registration)**		Medium	5.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
15	As a public user I want to browse books based on author, ISBN or book title by entering into the search field so that I can quickly find the book based on different search criteria.		Critical	5.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
16	As a public user I want to browse books based on categories by selecting a dropdown menu so that I can quickly find the book of my favourite categories.		Critical	5.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
17	As a public user I want to add a book to my cart so that I can keep track of the book I want to buy.		Critical	8.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.

18	As a public user I want to publish my used book to the website so that I can advertise my books to other users	Critical	3.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
19	As a public user I want to add a review to a user so that I can let other people know about the service the user provides.	Medium	8.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
20	As a public user I want to edit book details so that I can display the most up-to-date information I want for the book.	High	3.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
21	As a public user I want to receive notification when someone makes a purchase of my published books so that I can take actions such as delivering the books to the buyer.	Critical	8.00	To Do	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
22	As a public user I want to add a book to my cart and buy books so that I don't have to go to the bookstore to buy books.	Critical	8.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
23	As a public user, I want to register my account so that I can start using the system.	Critical	5.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
24	**As an admin/shop owner/public user, I want to see my profile information so that I can confirm that all of my information is correct.**	Critical	5.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
25	As a public user, I want to login my account so that I can start using the system.	Critical	5.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
26	As a public user, I want to see the contact us page so that I can know how to contact and give feedback	Low	3.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.
27	As a public user, I want to see the about us page so that I can know more information about the platform	Low	3.00	Done	Unit tests have been written and passed. Code is well formatted and suitable code comments are provided. Acceptance criteria for the User Stories is met. Signed off by Product Owner. Code review is completed.

SPRINT PLANNING

Sprint 3

1.Goal

- Allow Admin to approve shop owners.
- Allow users/shop owners to access their transaction history.

2. Duration of the sprint

2 weeks

3. What is the team's vision for this sprint?

- *The items committed are 6, 7, 22 because these are CRUD operations any users will need this functionality (for example adding a book to the cart and buying books).*
- *Potentially shippable product:*
- *Admin can approve shop owners so that only legal users can use the platform.*
- *Users/shop owners can access their transaction history.*

4. Estimation in story points

1. **User story 7**

   As an Admin I want to be able to approve shop owners so that I can allow only legal users to be members.

   Estimation point: 5

   Justification: Need to implement a react page to approve shop owners and an API (GET request) to get all the pending shop owners and an API (PUT request) to approve the shop owners.

2. **User story 6**

   As a public user/shop owner, I want to see my transaction history so that I can keep track of the books I have purchased or sold.

   Estimation point: 5

Justification: Need to implement a react page to display the transaction history and an API (GET request) for the books purchased and sold.

Sprint 4

1. Goal

- Finish documentation
- Successful deployment of the Frontend and Backend.
- Refactoring code.
- Write more unit tests
- Allow users to add books to the cart and purchase them online.

2. Duration of the sprint

2 weeks

3. What is the team's vision for this sprint?

- *Potentially shippable product:*
- *Complete documentation and making sure that we haven't missed any necessary documents*
- *Making sure that the our code is readable and has meaningful comments*
- *Writing unit tests to make sure that our product does not crash unexpectedly and behave weirdly.*
- *Users can add a book to their cart and checkout.*

4.Estimation in story points

1. User story 22

As a public user, I want to add a book to my cart and make the payment online using my credit card so that I don't have to go to the bookstore to buy books.

Estimation point: 8

Justification: Need to implement a react page for users to add their books and an API (POST request) to buy the books.

SPRINT REVIEW

Sprint 3

Sprint Goal

- The initial goal of this sprint was to finish documentation, finish writing up unit tests to test both frontend and backend, refactor code and successfully deploy the frontend and the backend of our product.

Completed Tasks

- We were able to complete all the tasks we were assigned, that includes: unit tests, refactoring and the one user stories.

What went well?

- Completed all the user stories that we planned.
- Maintained communication with team members
- Was able to avoid mistakes we made in the earlier sprints

What problems were faced and how were they resolved?

- Sprint 3 went pretty well, probably we underestimate our sprint velocity a little, however, we manage to complete all the critical user stories.

Next Sprint 4 start date:

- 1/10/21

Sprint 4

Sprint Goal

- The initial goal of this sprint was to finish documentation, finish writing up unit tests to test both frontend and backend, refactor code and successfully deploy the frontend and the backend of our product.

Completed Tasks

- We were able to complete all the tasks we were assigned, that includes; documentation, deployment, unit tests, refactoring and the two user stories.

What went well?

- Completed all the user stories that we planned.
- Maintained communication with team members
- Was able to avoid mistakes we made in the earlier sprints

What problems were faced and how were they resolved?

- We had some problems with deployment of the APIs and integrating backend and frontend, so we did some research on the internet and asked our tutor/lecturer to resolve the problem.

SPRINT RETRO

Sprint 3:

Bookeroo Sprint Retro Notes

Team: Team 5

Sprint: 3

Date: 04/10/21

Attended: Tuan Vu, Phu Nguyen, Janidu Higgoda, Edward Kahiro Kuo

Scrum Master: Tuan Vu

Product Owner: Mohamad Ali

Development team: Tuan Vu, Phu Nguyen, Janidu Higgoda, Edward Kahiro Kuo

1. Things That Went Well

- Complete a user story and avoid the mistake that we made in Sprint 1.

2. Things That Could Have Gone Better

- We could have completed more user stories, (we underestimate the sprint team velocity).
- Making less CI fails.

3. Things That Surprised Us

- How fast time flies by.
- Complexity in dockerisation.

4. Lessons Learned

- Keeping a steady development pace is important.
- Keep naming convention consistent between microservices.

5. Final Thoughts

- We have learnt our mistakes in Sprint one and improve, we avoid underestimating the complexity of user stories.

Sprint 4:

Bookeroo Sprint Retro Notes

Team: Team 5

Sprint: 4

Date: 22/10/21

Attended: Tuan Vu, Phu Nguyen, Janidu Higgoda, Edward Kahiro Kuo

Scrum Master: Tuan Vu

Product Owner: Mohamad Ali

Development team: Tuan Vu, Phu Nguyen, Janidu Higgoda, Edward Kahiro Kuo

1. Things That Went Well

- Deployment of FE and BE to AWS.

- Finish the last two user stories.

2. Things That Could Have Gone Better

- Setting up our database on RDS
- Free tier in AWS

3. Things That Surprised Us

- IP being dynamic in ECS tasks
- Target groups in Load Balancer being confusing
- AWS costing more than expected

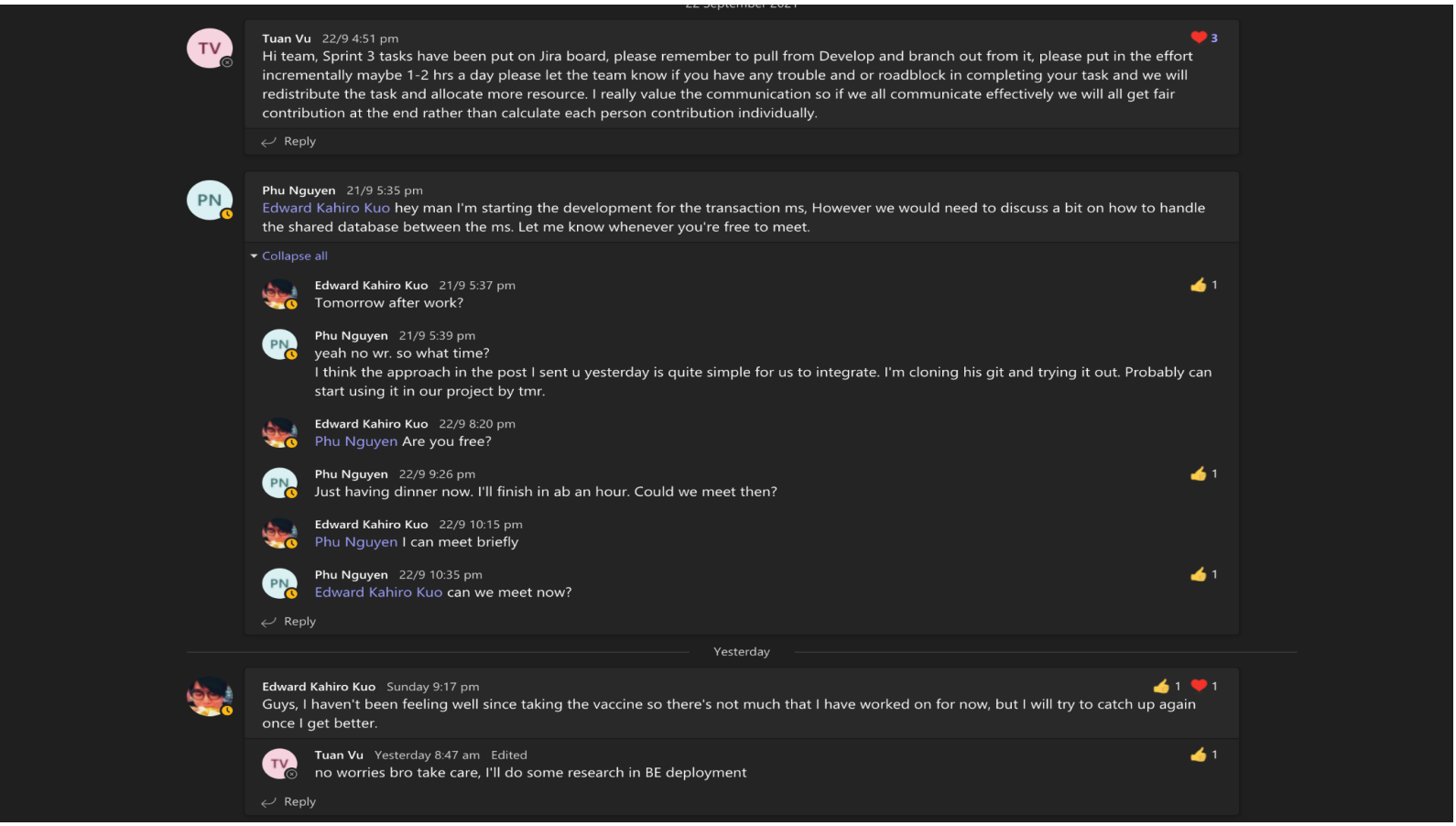
4. Lessons Learned

- Learn to communicate effectively.
- Estimate sprint velocity more accurately.

5. Final Thoughts

- We have learnt our mistakes and improved our performance over each sprint.

COMMUNICATION LOG



22 September 2021

Tuan Vu 22/9 4:51 pm — ❤️ 3

Hi team, Sprint 3 tasks have been put on Jira board, please remember to pull from Develop and branch out from it, please put in the effort incrementally maybe 1-2 hrs a day please let the team know if you have any trouble and or roadblock in completing your task and we will redistribute the task and allocate more resource. I really value the communication so if we all communicate effectively we will all get fair contribution at the end rather than calculate each person contribution individually.

Reply

Phu Nguyen 21/9 5:35 pm

Edward Kahiro Kuo hey man I'm starting the development for the transaction ms, However we would need to discuss a bit on how to handle the shared database between the ms. Let me know whenever you're free to meet.

Collapse all

Edward Kahiro Kuo 21/9 5:37 pm — 👍 1

Tomorrow after work?

Phu Nguyen 21/9 5:39 pm

yeah no wr. so what time?
I think the approach in the post I sent u yesterday is quite simple for us to integrate. I'm cloning his git and trying it out. Probably can start using it in our project by tmr.

Edward Kahiro Kuo 22/9 8:20 pm

Phu Nguyen Are you free?

Phu Nguyen 22/9 9:26 pm — 👍 1

Just having dinner now. I'll finish in ab an hour. Could we meet then?

Edward Kahiro Kuo 22/9 10:15 pm

Phu Nguyen I can meet briefly

Phu Nguyen 22/9 10:35 pm — 👍 1

Edward Kahiro Kuo can we meet now?

Reply

Yesterday

Edward Kahiro Kuo Sunday 9:17 pm — 👍 1 ❤️ 1

Guys, I haven't been feeling well since taking the vaccine so there's not much that I have worked on for now, but I will try to catch up again once I get better.

Tuan Vu Yesterday 8:47 am Edited — 👍 1

no worries bro take care, I'll do some research in BE deployment

Reply

Janidu Higgoda 24/9 7:34 pm
i have implemented the shopping cart page and order history page. i have also rendered the navbar for the shopping cart icon to be displayed upon login.

❤ 1

Reply

Tuan Vu 24/9 7:36 pm
Janidu Higgoda can you make PRs into Develop branch and ask for reviewer ?

Collapse all

Janidu Higgoda 24/9 7:37 pm
yep, i will do that now

Tuan Vu 24/9 7:37 pm
thanks bro

Reply

Tuan Vu 23/9 8:36 pm
Hi backend team, I also need the endpoint to get all the pending shop owners please implement and fill in the docs, Phu Nguyen Edward Kahiro Kuo

Collapse all

Edward Kahiro Kuo 23/9 8:42 pm
Hi, it hasn't been implemented yet. I'll create a ticket for it.

Tuan Vu 23/9 8:43 pm
all good thanks Edward

Phu Nguyen 24/9 5:50 pm
Tuan Vu this backend endpoint is done and I added the GET request on the integration doc.

Edward Kahiro Kuo 24/9 7:06 pm
Phu Nguyen Thanks Phu.

❤ 1

Tuan Vu 24/9 7:16 pm
awesome thanks Phu Nguyen

❤ 1

Edward Kahiro Kuo 24/9 7:28 pm
I think I might need to create a new email or turn off email notifications cause my inbox is clogged with github right now.

😂 1

Phu Nguyen 24/9 7:29 pm
I think you can filter all github mails to a folder in your inbox automatically

😬 1

Phu Nguyen 24/9 7:35 pm
Edward Kahiro Kuo go to your outlook on web, click on setting wheel on top right, then "View all Outlook settings" on the bottom.

See more

Tuan Vu 24/9 7:40 pm
hey Phu Nguyen you forgot to add what the endpoint returns

Phu Nguyen 24/9 8:26 pm
Tuan Vu oh yeah. it would be exactly the same as getAllUser

Edward Kahiro Kuo 24/9 9:39 pm
I'm not using outlook for github 😕
But I'll try to find a way

Reply

---

30 September 2021

Tuan Vu 30/9 11:28 pm
Hi team, I have deployed the frontend using docker link here: http://frontend-1990882630.ap-southeast-2.elb.amazonaws.com/login

Bookeroo: Built by book lovers, for book lovers
Web site created using create-react-app
frontend-1990882630.ap-southeast-2.elb.amazonaws.com

Tuan Vu 30/9 11:30 pm
Also, Phu Nguyen Edward Kahiro Kuo **Janidu Higgoda** please approve my PRs when you get a chance

Reply

Yesterday

Tuan Vu Yesterday 12:09 pm
Hi team please make commit incrementally in preparation for monday progress mark

👍 1

Reply

10 October 2021

Phu Nguyen 6/10 7:10 pm
Tuan Vu all the GET requests endpoint are fixed and added, I've put the JSON return on the integration doc so you can check if the return info is sufficient. Still need to fix CI, add security, data validation and unit tests, but the basic functionality is done.

Collapse all

Tuan Vu 6/10 7:37 pm
awesome thanks Phu I'll try to do the integration
👍 1

Tuan Vu 8/10 4:49 pm
Hey Phu Nguyen
Can you also add the books information for each order ? I will need the book title, book author, book image and short description
👍 1

Phu Nguyen 8/10 4:51 pm
ok sure.
👍 1

Phu Nguyen Sunday 5:51 pm
Tuan Vu alright I got the additional fields in. Updated the integration doc. Should be good to go for tonight integration.

Tuan Vu Sunday 7:19 pm
great thanks Phu Nguyen 😊

Reply

Today

Tuan Vu 10:00 am
hi team, tomorrow we will integrate all the features into develop, Phu Nguyen have you got a chance to fix the CI for transaction MS ?

Phu Nguyen 10:12 am
been so busy with the CT A2 T.T but I'll take today to get it done. We should be good to go for fully integration tmr.
👍 1

Reply

Tuan Vu 5:56 pm
Hi team I have put up the slide template on our drive please fill your part when you have a chance

Reply

Tuan Vu 9:01 pm
Hi team, let me know if you guys are free to meet tomorrow at 11:00 AM
👍 2

Reply

Phu Nguyen 3/10 1:54 am
Edward Kahiro Kuo ping me whenever you have some free time. I need your help troubleshooting some issue 😬

Collapse all

Edward Kahiro Kuo 3/10 7:24 am
Sorry guys but I won't be on much since my sickness reappeared, I'll try to help out when I can. I saw a ticket assigned to you regarding backend database, are you currently doing it? I'm currently working on the database access for the docker.

Phu Nguyen 3/10 11:03 am
Edward Kahiro Kuo the database is just like we discussed that we still use the other ms as dependency so that ticket is basically finished. I'm having issue calling the API from postman, keep getting 404 so if you could help debug then that would be great.

Phu Nguyen 3/10 11:16 am
lmao just fixed it. no worries Edward Kahiro Kuo.
👍 1

Edward Kahiro Kuo 3/10 7:17 pm
Sorry I meant the dockerisation of the database

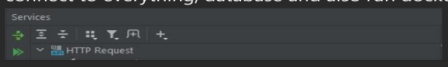
Phu Nguyen 3/10 7:50 pm
Edward Kahiro Kuo yeah I havent working on that yet so please do

Reply

Edward Kahiro Kuo Thursday 4:57 pm

❤️ 1

I've deployed our DB to RDS using aurora, configuring docker for deployment and ECS now. Also, another handy function on IntelliJ, you can connect to everything, database and also run docker on it.



See more

Collapse all

Edward Kahiro Kuo Thursday 8:25 pm
Did anyone change the docker settings at all? Phu Nguyen

Phu Nguyen Thursday 8:52 pm
Edward Kahiro Kuo hmm I didnt touch it. Is there anything wrong?

Edward Kahiro Kuo Thursday 9:27 pm
Phu Nguyen for some reason the docker is all broken

Phu Nguyen Thursday 9:28 pm
Edward Kahiro Kuo you want to meet now and fix it together?

Edward Kahiro Kuo Thursday 9:29 pm
Sure

👍 ❤️ 😂 😮 😢 😕 | ⋯

Phu Nguyen Thursday 9:29 pm
Alright see you in Discord channel. I'll join u in a sec.

Tuan Vu Yesterday 9:14 am
Great stuff Edward Kahiro Kuo 🙂 rn please spin up and test the api, you can disable security for now, and let me know the domain of the API for loginms, bookms, and transaction ms

Edward Kahiro Kuo Yesterday 11:57 am

❤️ 1

The docker is fixed and integrated with DB, now I'm figuring out how to deploy.

Reply

Today

Edward Kahiro Kuo 9:36 pm

👍 1

Phu Nguyen Can you give me admin access to the github repo? I need it for setting up the automated deployment.

Collapse all

Phu Nguyen 9:41 pm
Edward Kahiro Kuo I can only see Collaborator user type. nowhere to set you as admin. Any clue on how could I do that?

Edward Kahiro Kuo 9:44 pm
Is there manager or something?

Reply

17 October 2021

Tuan Vu 16/10 9:01 pm
Hi team, let me know if you guys are free to meet tomorrow at 11:00 AM

👍 2

Collapse all

Phu Nguyen Sunday 10:29 am
Tuan Vu Edward Kahiro Kuo **Janidu Higgoda** So srr guys I only see this now but could we move to tonight around 8pm?

👍 1

This message has been deleted.

Tuan Vu Sunday 10:30 am
Im cool with it

👍 1

Edward Kahiro Kuo Sunday 10:31 am
Tuan Vu I'm going to be getting my second dose this afternoon so I'm not sure if I will be well enough to join at 8, can you hop on at 11 for a bit so I can show you the deployment stuff?

Tuan Vu Sunday 10:32 am
yeh sure no problem I can meet you at 11

Reply

18 October 2021

Tuan Vu Sunday 9:11 pm
Edward Kahiro Kuo can you share your AWS credentials let me see if I can fix it

Collapse all

Edward Kahiro Kuo Monday 9:26 am
Tuan Vu Is this the cors issue? Also how did the integration go yesterday for the transactions?

Tuan Vu Monday 11:31 am
yes the cors issue but its strange cause I can fetch it from api/books but not getting particular books so there maybe some blockage in the setup, I did not continue with the integration yesterday I'll check it today and let you guys know

👍 1

Edward Kahiro Kuo Monday 11:32 am
I'm thinking of creating the cluster for the transaction service today so we can deploy everything at once.

👍 1

Reply

Tuan Vu Monday 5:02 pm
hey Edward Kahiro Kuo , I tried to test the backend on localhost but the database has not been seeded,

```
server.port=8080

# Local SQL configuration
spring.datasource.url=jdbc:mysql://localhost:3306/bookeroo
spring.datasource.username=root
```

See more

Collapse all

Edward Kahiro Kuo Monday 6:18 pm
Tuan Vu Is the properties in transaction set to update?

Tuan Vu You can check if the configuration files are commented out. You can uncomment the files.

Reply

Tuan Vu Monday 7:26 pm
hi team please progress with the documentation the deadline for documentation is Friday night

Reply

Tuan Vu Monday 7:44 pm
hi team, I cannot test the backend on my localhost with the branch Develop. we will need to meet on Wednesday to the integration again, either Phu Nguyen or Edward Kahiro Kuo please check the backend code

Collapse all

Edward Kahiro Kuo Monday 7:44 pm
Tuan Vu what's wrong with it?

Tuan Vu Monday 7:45 pm
the seeding data doesn't work, I tried register a new user but it doesn't work too

Edward Kahiro Kuo Monday 7:47 pm
Tuan Vu Did you try the usual login->book->transaction

Tuan Vu I think I know what's causing this. Can you try changing the ddl auto to 'create' in the application properties. And comment out the * in the configuration files in book transaction.

So all 3 microservices have create ddl auto.

Tuan Vu Monday 7:54 pm
I got this error when I comment out the user configuration file

```
@Bean
CommandLineRunner seedUsers(UserRepository userRepository, RoleRepository roleRepository){
    return args -> {
        Role adminRole = Role.builder().code("
        Role userRole = Role.builder().code("ROLE_PUBLIC").name("Public User").build();
```

Could not autowire. No beans of 'UserRepository' type found.

Edward Kahiro Kuo Monday 7:54 pm
Tuan Vu try running it anyway

Tuan Vu Monday 7:56 pm
all good now thanks bro !!

👍 2

Reply

19 October 2021

Tuan Vu Tuesday 11:55 am
Hi team, we will integrate docs on Friday and do the presentation rehearsal on Sunday this week please prepare for it, Phu Nguyen **Janidu Higgoda** I know its stressful as many Assignment is due, please let me know if you need me with OSP I'm happy to help

👍 1 ❤️ 1

Reply

Edward Kahiro Kuo Wednesday 7:37 pm
So guys, my services on aws got terminated because of billing. Apparently I chose the wrong options for rds and fargate causing my bill to blow up so I had to remove everything but I'm setting it all back up tonight. Tuan Vu what did you choose for the ecs cluster and service? network + linux?

Collapse all

Tuan Vu Wednesday 8:03 pm
I'm using fargate, this cluster template, it costs me around 11 dollars now, I think we only need it for the demo and testing, so you can spin up, test and then turn off then on demo day turn it on, you can take some screenshots in case we cannot turn it on for demo day. The bill I think we should divide equally in the end Phu Nguyen **Janidu Higgoda**

See more

👍 2 ❤️ 1

Edward Kahiro Kuo Wednesday 9:12 pm
Ok, I've redeployed everything. The services should be running perfectly on AWS now.

Tuan Vu Phu Nguyen **Janidu Higgoda** Can you guys approve the PR so we can automate all deployments
https://github.com/s3516565/SEPT_TeamProject_Sem2_2021/pull/100

Tuan Vu Wednesday 9:16 pm
the IP is the same Edward Kahiro Kuo ?

Edward Kahiro Kuo Wednesday 9:18 pm
Tuan Vu yes. The IP is an elastic IP mapped to the load balancer.

Reply

Tuan Vu Wednesday 9:04 am
hey Edward Kahiro Kuo please let me know how it goes with the backend deployment I may need to meet with you today or tmr to complete the ci/cd

👍 1

Collapse all

Edward Kahiro Kuo Wednesday 9:20 pm
I didn't see your message. Did you still need to meet? I'm free after work tomorrow. I will be doing documentations tomorrow.

Tuan Vu Wednesday 9:21 pm
no worries all good 😊

Reply

Tuan Vu Monday 8:53 pm

hey man Edward Kahiro Kuo look like its the error with how we set up, I tried testing it with Postman, and it did not work, maybe you can fix it from Postman first

See more

Collapse all

Edward Kahiro Kuo Tuesday 11:49 am

There's something weird going on with the database integration. I'm checking it.

Edward Kahiro Kuo Tuesday 1:37 pm

Tuan Vu Found the fix. I will be deploying it later today along with the CD docker for transaction.

Reply

20 October 2021

Tuan Vu Wednesday 9:05 am

here's the fe deployed with ecs http://sept-fe-alb-1073804615.ap-southeast-2.elb.amazonaws.com/login

Bookeroo: Built by book lovers, for book lovers
Web site created using create-react-app

sept-fe-alb-1073804615.ap-southeast-2.elb.amazonaws.com

Reply

Tuan Vu Wednesday 3:37 pm

hey Phu Nguyen please add me as github repo admin I need to set up CD

Collapse all

Edward Kahiro Kuo Wednesday 3:57 pm

Last time we tried but we couldn't. Admins are only for github enterprise.

Phu Nguyen Wednesday 4:01 pm

Tuan Vu ey last time we tried but sady Github does not let me. If u need to add anything, please let me know and I'll them for you asap.

Tuan Vu Wednesday 4:05 pm Edited

can you please add AWS access key to github actions, you can name the variable whatever you like and let me know , thanks:
AWS access key ID: AKIAXJFSENGNBC3LZO6E
AWS secret: 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

Tuan Vu Wednesday 6:34 pm

Phu Nguyen how it's going ?

Phu Nguyen Wednesday 7:07 pm

Tuan Vu Monday 5:02 pm

hey Edward Kahiro Kuo , I tried to test the backend on localhost but the database has not been seeded,

```
server.port=8080

# Local SQL configuration
spring.datasource.url=jdbc:mysql://localhost:3306/bookeroo
spring.datasource.username=root
```

See more

Collapse all

Edward Kahiro Kuo Monday 6:18 pm

Tuan Vu Is the properties in transaction set to update?

Tuan Vu You can check if the configuration files are commented out. You can uncomment the files.

Reply

Tuan Vu Monday 7:26 pm

hi team please progress with the documentation the deadline for documentation is Friday night

Reply

Tuan Vu Wednesday 10:29 pm

3

Great news guys Edward Kahiro Kuo Phu Nguyen **Janidu Higgoda** the site is up and frontend/backend CD have been set up, kudos for u Edward Kahiro Kuo nice work with the deployment !!, please approve PRs, its all working well after I tested it, http://sept-fe-alb-1073804615.ap-southeast-2.elb.amazonaws.com/login

Bookeroo: Built by book lovers, for book lovers
Web site created using create-react-app

sept-fe-alb-1073804615.ap-southeast-2.elb.amazonaws.com

Collapse all

Edward Kahiro Kuo Wednesday 11:10 pm

Thanks Tuan Vu, you too! And everyone else! I've approved the PRs and merged the relevant changes, so we're all good now.

Edward Kahiro Kuo Wednesday 11:22 pm

I found some errors on the website, not sure if this was intended. The address and phone doesn't change when you change on the profile menu. Also, when inputting invalid ISBN, it doesn't save, but it doesn't tell you anything.

Tuan Vu Wednesday 11:23 pm

1

I see lets me fix that

Edward Kahiro Kuo Wednesday 11:24 pm

Also, there is a small problem with the display which I think might be caused by the backend aws latency, but I don't think there's anything we can do with that except refreshing the page. The ecs also boots up a bit slow when it starts after being redeployed.

Tuan Vu Wednesday 11:26 pm

1

profile change has not been implemented I think, but I'll add more custom error box

Edward Kahiro Kuo Wednesday 11:27 pm

Thanks.

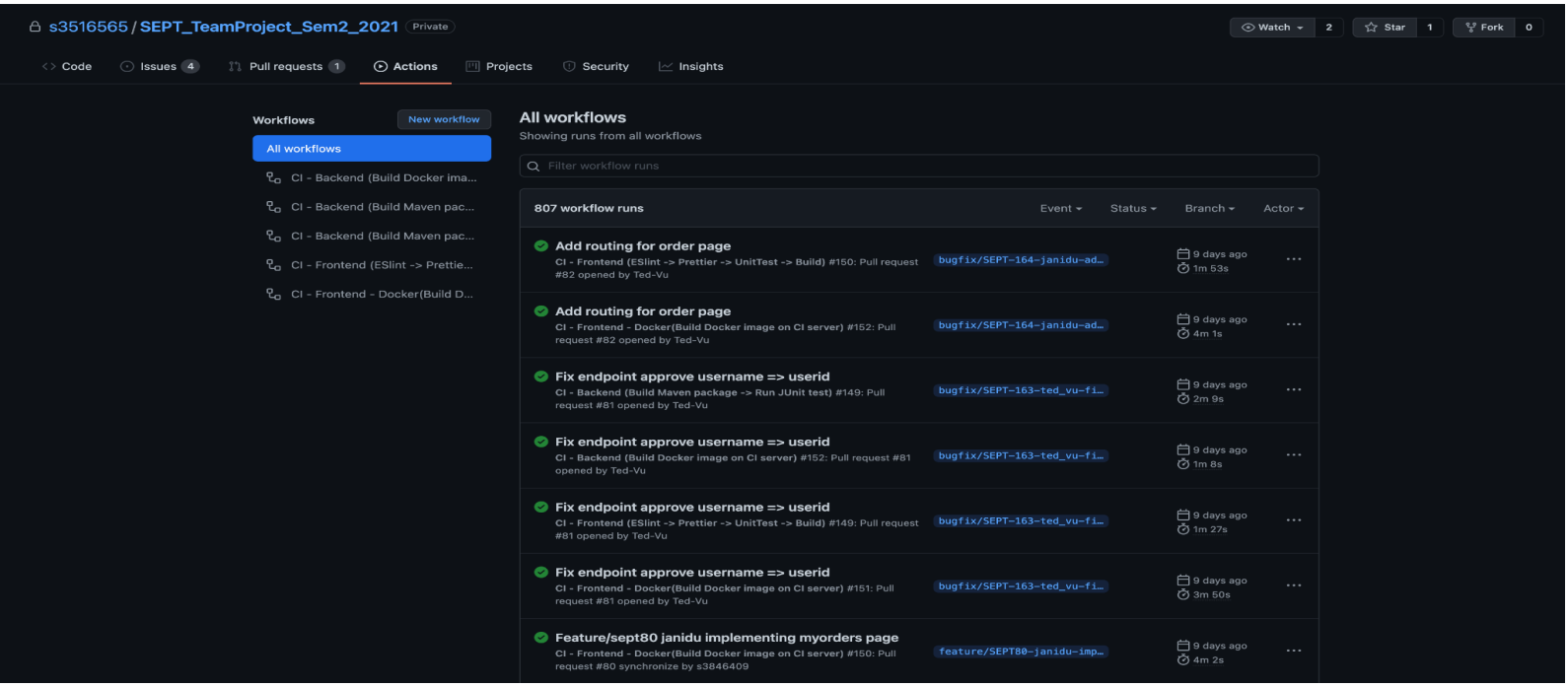
Wednesday 11:28 pm

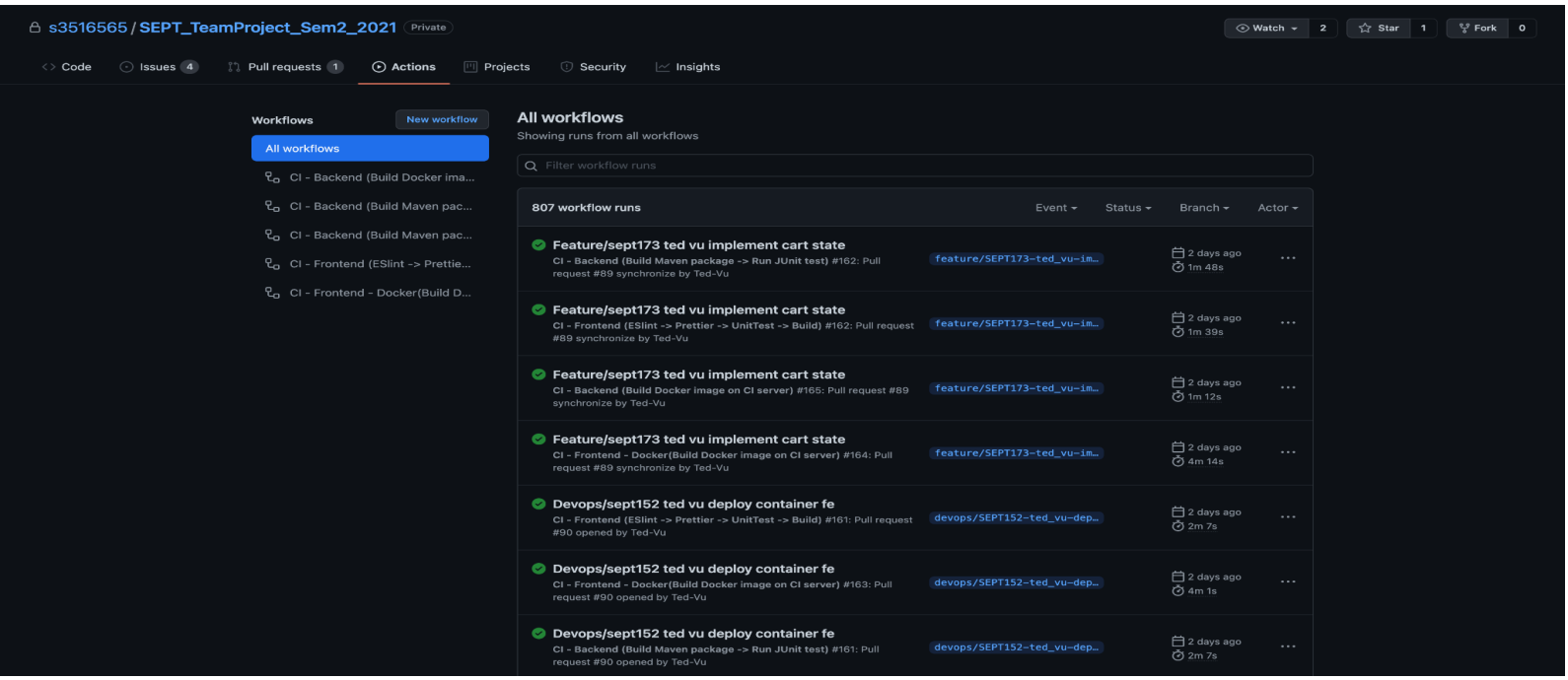
1

Hopefully it's not too much trouble.

Reply

CI SCREENSHOTS FROM GITHUB ACTION





Change static page shopping cart CI - Backend (Build Maven package -> Run JUnit test) #154: Pull request #86 opened by Ted-Vu	bugfix/SEPT171-ted_vu-bug…	6 days ago 2m 3s	•••
Change static page shopping cart CI - Frontend (ESlint -> Prettier -> UnitTest -> Build) #154: Pull request #86 opened by Ted-Vu	bugfix/SEPT171-ted_vu-bug…	6 days ago 1m 38s	•••
Change static page shopping cart CI - Backend (Build Docker image on CI server) #157: Pull request #86 opened by Ted-Vu	bugfix/SEPT171-ted_vu-bug…	6 days ago 1m 2s	•••
Change static page shopping cart CI - Frontend - Docker(Build Docker image on CI server) #156: Pull request #86 opened by Ted-Vu	bugfix/SEPT171-ted_vu-bug…	6 days ago 3m 36s	•••
fix routing to books CI - Frontend - Docker(Build Docker image on CI server) #155: Pull request #85 opened by s3846409	bugfix/SEPT82-janidu-shop…	8 days ago 3m 42s	•••
fix routing to books CI - Frontend (ESlint -> Prettier -> UnitTest -> Build) #153: Pull request #85 opened by s3846409	bugfix/SEPT82-janidu-shop…	8 days ago 2m 1s	•••
fix routing to books CI - Backend (Build Maven package -> Run JUnit test) #153: Pull request #85 opened by s3846409	bugfix/SEPT82-janidu-shop…	8 days ago 2m 6s	•••
fix routing to books CI - Backend (Build Docker image on CI server) #156: Pull request #85 opened by s3846409	bugfix/SEPT82-janidu-shop…	8 days ago 1m 0s	•••
Feature/sept81 janidu add notification to shoppingca… CI - Backend (Build Docker image on CI server) #155: Pull request #84 opened by s3846409	feature/SEPT81-janidu-add…	8 days ago 1m 11s	•••
Feature/sept81 janidu add notification to shoppingca… CI - Frontend (ESlint -> Prettier -> UnitTest -> Build) #152: Pull request #84 opened by s3846409	feature/SEPT81-janidu-add…	8 days ago 2m 5s	•••
Feature/sept81 janidu add notification to shoppingca… CI - Frontend - Docker(Build Docker image on CI server) #154: Pull request #84 opened by s3846409	feature/SEPT81-janidu-add…	8 days ago 3m 33s	•••

Run duration

MEETING MINUTES

Date	Venue	Attendees	Action Item	Information/decision
Monday, 20/09/2021	Microsoft Teams	Phu Nguyen	_ Backend endpoints (Sunday 26/09)	_ Decide user stories for sprint 3 _ Database changes to accommodate the new user stories
		Tuan Vu	_ Frontend: Admin approve shop owner user _ Front end dev-ops _ Front end integration	
		Edward Kahiro Kuo	_ Backend DevOps	
		Janidu Higgoda	_ Frontend components: Shopping cart, Navbar, profile page shows transaction (Friday 24/09)	
Friday, 29/09/2021	Microsoft Teams	Phu Nguyen	_ Backend endpoints for order book	_ Integration to get all the shop owners pending endpoint _ Merge pull requests
		Tuan Vu	_ Stage management for transactions _ Prepare integration for transaction page	
		Edward Kahiro Kuo	_ Deployment + dockerization	
		Janidu Higgoda	_ Fix routing for shopping cart page. _ Fix Navbar, displaying the number of books	

			on shopping cart icon.	
Monday, 27/09/2021	Collab	Phu Nguyen	_ Backend endpoints for order book	
		Tuan Vu	_ Stage management for transactions _ Prepare integration for transaction page	
		Edward Kahiro Kuo	_ Deployment + dockerization of the microservices	
		Janidu Higgoda	_ Testing _ Documentation	
Friday, 01/10/2021	Microsoft Teams	Phu Nguyen	_ Backend endpoints for order book	
		Tuan Vu	_ Testing	
		Edward Kahiro Kuo	_ Deployment + dockerization of the microservices	
		Janidu Higgoda	_ Testing _ Documentation	
Sunday, 03/10/2021	Microsoft Teams	Phu Nguyen	_ Backend endpoints for order book	
		Tuan Vu	_ Testing	_ Integration for buying books endpoint _ Merge pull requests
		Edward Kahiro Kuo	_ Deployment + dockerization of the microservices	

		Janidu Higgoda	_ Testing _ Documentation	
Monday, 04/10/2021	Collab	Phu Nguyen	_ Backend endpoints for order book _documentation(acceptance tests)	
		Tuan Vu	_ refactoring code and writing the refactoring report	
		Edward Kahiro Kuo	_ Deployment + dockerization of the microservices _documentation(sprint retro)	
		Janidu Higgoda	_ Testing _ Documentation	
Sunday, 10/10/2021	Microsoft Teams	Phu Nguyen	_ fix minor errors _documentation(acceptance tests)	_ Integration for my orders endpoint and my sold books endpoint _ Merge pull requests
		Tuan Vu	_ refactoring code and writing the refactoring report	
		Edward Kahiro Kuo	_ Deployment + dockerization of the microservices _documentation(sprint retro)	

		Janidu Higgoda	_ Testing _ Documentation	
Monday, 11/10/2021	Collab	Phu Nguyen	_documentation(acceptance tests, System architecture diagram, Scrum process description) _prepare presentation slides	
		Tuan Vu	_ documentation(Vision statement, Deployment diagram for FE, refactoring code and writing the refactoring report _prepare presentation slides	
		Edward Kahiro Kuo	_ Deployment + dockerization of the microservices and the database _documentation(sprint retro, Gitflow overview and organisation, Deployment diagram for BE) _prepare presentation slides	
		Janidu Higgoda	_ Testing _ Documentation(Meeting minutes, Sprint planning,	

			Communication log) _prepare presentation slides	
Monday, 17/10/2021	Microsoft Teams	Phu Nguyen	_documentation(acceptance tests, System architecture diagram, Scrum process description)	_Merge pull requests
		Tuan Vu	_documentation(Vision statement, Deployment diagram for FE, refactoring code and writing the refactoring report	
		Edward Kahiro Kuo	_documentation(sprint retro, Gitflow overview and organisation, Deployment diagram for BE) _prepare presentation slides	
		Janidu Higgoda	_ Testing _ Documentation(Meeting minutes, Sprint planning, Communication log)	
Friday, 22/10/2021	Microsoft Teams	Phu Nguyen	_documentation(acceptance tests, Scrum process description)	
		Tuan Vu	_documentation(refactoring code	

			and writing the refactoring report	_Documentation
		Edward Kahiro Kuo	_documentation(sprint retro, Deployment diagram for BE) _prepare presentation slides	
		Janidu Higgoda	_Documentation(Meeting minutes, product backlog)	

REFACTORING REPORT

Frontend:

1. **Use React html properties keyword**
   Some properties keywords in HTML generated warnings with React were omitted such as class and for. These keywords are replaced with className and htmlFor respectively.

   Use className instead of class to avoid warnings

```
<div className="order-item" key={index}>
  {" "}
  <hr></hr>
```

   Use htmlFor instead of for to avoid warnings

```
<label htmlFor="long description">Long Description</label>
<textarea
```

2. **Remove warning by adding comment “eslint-disable-next-line react-hooks/exhaustive-deps”**

```
useEffect(() => {
  window.scrollTo(0, 0);
  searchViewBook(location.state.bookID, history);
  // eslint-disable-next-line react-hooks/exhaustive-deps
}, []);
```

3. **Put all endpoints into a single file.**

```
export const bookEndpoint = "http://localhost:8090/api/";
export const userEndpoint = "http://localhost:8080/api/";
export const transactionEndpoint = "http://localhost:8070/api/";
```

4. **Format with npm prettier**

```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test -u",
  "eject": "react-scripts eject",
  "deploy:dev": "aws s3 sync build s3://sept-ted-vu-static-hosting",
  "lint:check": "eslint --fix \"./**/src/**/*.{js,jsx}\"",
  "format:check": "prettier --check \"./**/src/**/*.{js,jsx}\"",
  "lint:fix": "eslint \"./**/src/**/*.{js,jsx}\"",
  "format:fix": "prettier --write \"./**/src/**/*.{js,jsx}\""
},
```

The code is formatted with npm prettier which is configured in the CI.

5. **Running lint check with ESLint.**

```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test -u",
  "eject": "react-scripts eject",
  "deploy:dev": "aws s3 sync build s3://sept-ted-vu-static-hosting",
  "lint:check": "eslint --fix \"./**/src/**/*.{js,jsx}\"",
  "format:check": "prettier --check \"./**/src/**/*.{js,jsx}\"",
  "lint:fix": "eslint \"./**/src/**/*.{js,jsx}\"",
  "format:fix": "prettier --write \"./**/src/**/*.{js,jsx}\""
},
```

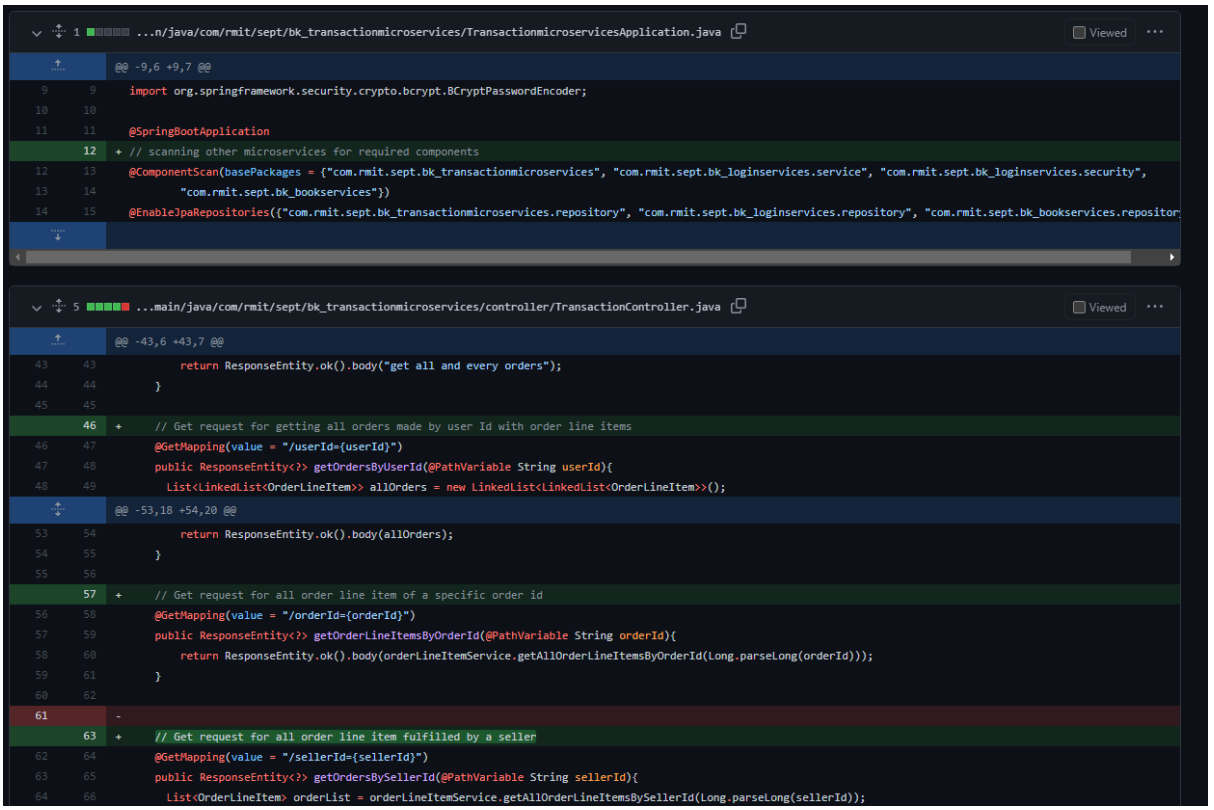
The code syntax is checked with ESLint.

6. **Refactor into functional components.**

   Components have been refactored to use functional components.

Backend:

1. **Add comments:**



...n/java/com/rmit/sept/bk_transactionmicroservices/TransactionmicroservicesApplication.java

```
@@ -9,6 +9,7 @@
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

@SpringBootApplication
+ // scanning other microservices for required components
@ComponentScan(basePackages = {"com.rmit.sept.bk_transactionmicroservices", "com.rmit.sept.bk_loginservices.service", "com.rmit.sept.bk_loginservices.security",
        "com.rmit.sept.bk_bookservices"})
@EnableJpaRepositories({"com.rmit.sept.bk_transactionmicroservices.repository", "com.rmit.sept.bk_loginservices.repository", "com.rmit.sept.bk_bookservices.repositor
```

...main/java/com/rmit/sept/bk_transactionmicroservices/controller/TransactionController.java

```
@@ -43,6 +43,7 @@
            return ResponseEntity.ok().body("get all and every orders");
        }

+    // Get request for getting all orders made by user Id with order line items
        @GetMapping(value = "/userId={userId}")
        public ResponseEntity<?> getOrdersByUserId(@PathVariable String userId){
          List<LinkedList<OrderLineItem>> allOrders = new LinkedList<LinkedList<OrderLineItem>>();
@@ -53,18 +54,20 @@
            return ResponseEntity.ok().body(allOrders);
        }

+    // Get request for all order line item of a specific order id
        @GetMapping(value = "/orderId={orderId}")
        public ResponseEntity<?> getOrderLineItemsByOrderId(@PathVariable String orderId){
            return ResponseEntity.ok().body(orderLineItemService.getAllOrderLineItemsByOrderId(Long.parseLong(orderId)));
        }

-
+    // Get request for all order line item fulfilled by a seller
        @GetMapping(value = "/sellerId={sellerId}")
        public ResponseEntity<?> getOrdersBySellerId(@PathVariable String sellerId){
          List<OrderLineItem> orderList = orderLineItemService.getAllOrderLineItemsBySellerId(Long.parseLong(sellerId));
```

2. **Changing naming convention for code readability:**

```
// get all order line items by a specific order id
public List<OrderLineItem> getAllOrderLineItemsByOrderId(long orderId){
    List<OrderLineItem> allOrderLineItems = orderLineItemRepository.findAll();
    List<OrderLineItem> orderLineItemsByOrderId = new LinkedList<OrderLineItem>();
    for (OrderLineItem orderLineItem : allOrderLineItems) {
        if (orderLineItem.getOrderId() == orderId) {
            orderLineItemsByOrderId.add(orderLineItem);
        }
    }
    return orderLineItemsByOrderId;
}

// get order line item by specific seller id
public List<OrderLineItem> getAllOrderLineItemsBySellerId(long sellerId){
    List<OrderLineItem> allOrderLineItems = orderLineItemRepository.findAll();
    List<OrderLineItem> orderLineItemsBySellerId = new LinkedList<OrderLineItem>();
    for (OrderLineItem orderLineItem : allOrderLineItems) {
        if (orderLineItem.getSellerId() == sellerId) {
            orderLineItemsBySellerId.add(orderLineItem);
        }
    }
    return orderLineItemsBySellerId;
}
```

Vision Statement

Our target customers:

- Public users: People who want to find their favorite books with a suitable price.
- Publisher: Popular publisher as well as self-publisher.

Our vision statement:

“To provide a platform for users to find books with suitable prices and incentivize self-publishers/novelists to bring their books to the audience.”

TESTING DOCUMENTATION

Testing Documentation

Type of testing conducted:

- Unit testing

- Integration testing
- Browser testing
- Mobile device compatibility testing
- Smoke testing
- Acceptance testing

1. Unit testing:

Frontend

- **Test location**: SEPT_TeamProject_Sem2_2021/FrontEnd/myfirstapp/src/test

- **Test strategy:**
  The full implementation of tests can be found in the above location here are some screenshots of the tests:

  o Test the rendering of each element

```
test("Render without crashing", () => {
  render(
    <BrowserRouter>
      <ShoppingCartPlain />
    </BrowserRouter>
  );
});
```

```
test("renders without crashing", () => {
  render(
    <BrowserRouter>
      <Card bookProps={book} />{" "}
    </BrowserRouter>
  );
});
```

- Test if element renders the props received correctly.

```
test("Render container element", () => {
  const { getByTestId } = render(
    <BrowserRouter>
      <ShoppingCartPlain />
    </BrowserRouter>
  );
  const element = getByTestId("container");
  expect(element).toBeInTheDocument();
});
```

```
test("renders card title element with correct content", () => {
  const { getByTestId } = render(
    <BrowserRouter>
      <Card bookProps={book} />{" "}
    </BrowserRouter>
  );
  const titleElement = getByTestId("card-title-element");
  expect(titleElement).toHaveTextContent(book.title);
});
```

- Snapshot testing (the snapshots are placed are in __snapshot folder and placed under the test folder)

```
test("matches snapshot", () => {
  const tree = renderer
    .create(
      <BrowserRouter>
        <ShoppingCartPlain />
      </BrowserRouter>
    )
    .toJSON();

  expect(tree).toMatchSnapshot();
});
```

```
// snapshot testing
test("matches snapshot", () => {
  const tree = ReactTestRenderer.create(
    <BrowserRouter>
      <Card bookProps={book} />{" "}
    </BrowserRouter>
  ).toJSON();

  expect(tree).toMatchSnapshot();
});
```

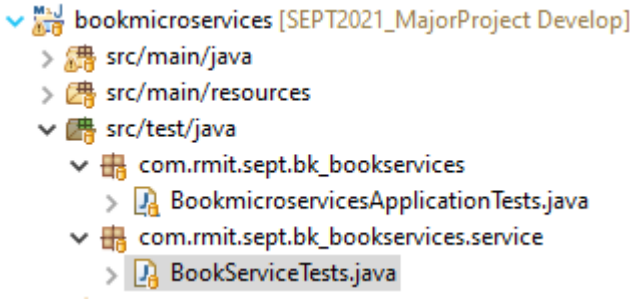
- **Test execution**

The unit test for Frontend consists of 18 test suites with 83 tests.

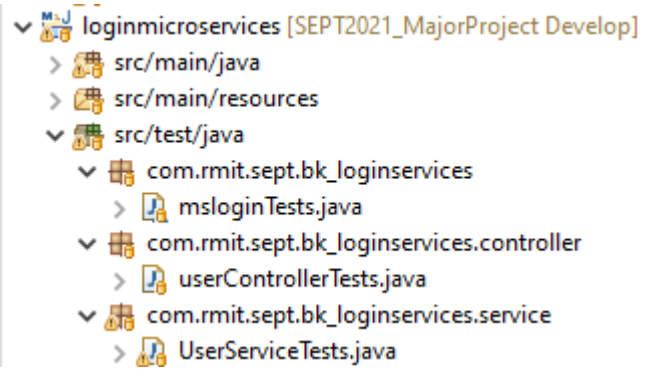
```
Test Suites: 18 passed, 18 total
Tests:       83 passed, 83 total
Snapshots:   3 written, 10 passed, 13 total
Time:        10.551s
Ran all test suites.

Watch Usage
```

Backend

- **Test location:**
  SEPT_TeamProject_Sem2_2021/BackEnd/loginmicroservices/src/test



SEPT_TeamProject_Sem2_2021/BackEnd/bookmicroservices/src/test



- **Test strategy**:
  The full implementation of tests can be found in the above location here are some screenshots of the tests:

  - Test for the method getBookByTitle to throw an exception when an invalid title was given.

```
@Test
void test_getBookByTitle_titleNotExist_fail(){
    doReturn(new ArrayList<Book>()).when(bookRepository).findByTitle(any());
    assertThrows(ItemNotFoundException.class, () -> {
        bookService.getBooksByTitle("invalidTitle");
    });
    verify(bookRepository, times(1)).findByTitle(any());
}
```

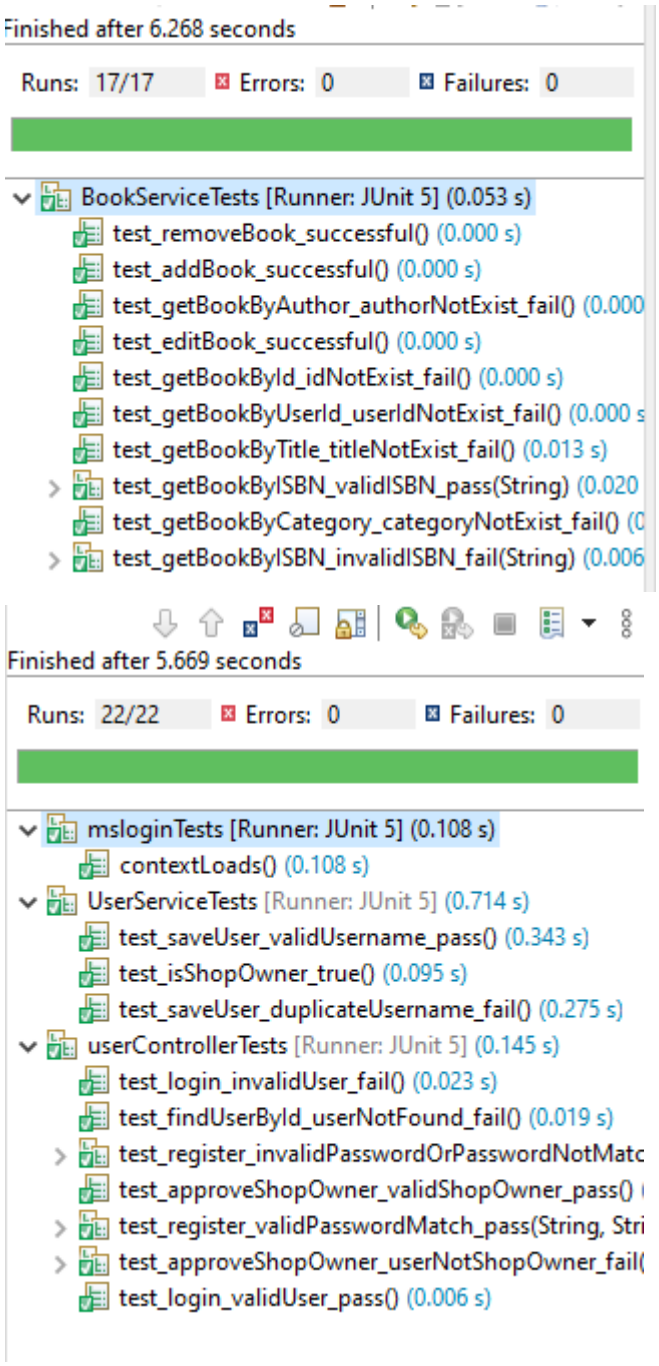
  - Test to check if the remove book method is working correctly:

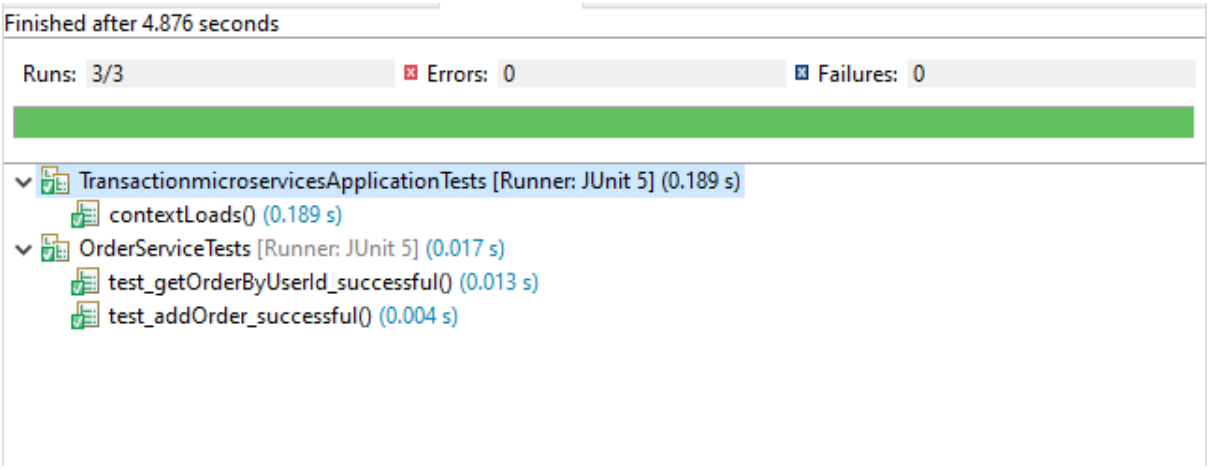
```
@Test
void test_removeBook_successful(){
    Book book = Book.builder()
            .userId((long)2)
            .title("Edited book Sword Art Online Progressive, Vol. 1")
            .publishedDate(LocalDate.of(2013, 10, 10))
            .shortDescription("The volume contains Kirito and Asuna's a
            .longDescription("There's no way to beat this game. The onl
            .ISBN10("0316259365")
            .coverImageUrl("https://d1w7fb2mkkr3kw.cloudfront.net/asset
            .price(100)
            .quantity(10)
            .status("New")
            .build();
    doNothing().when(bookRepository).delete(any());
    bookService.removeBook(book);
    verify(bookRepository, times(1)).delete(book);
}
```

- Test when create new user with duplicate username, it would throw the correct exception:

```
@Test
void test_saveUser_duplicateUsername_fail(){
    User user = User.builder().username("test_usernameExists@rmit.com").password("password")
            .givenName("given").surname("sur").build();
    User duplicateUser = User.builder().username("test_usernameExists@rmit.com").password("password")
            .givenName("given").surname("sur").build();
    userRepository.save(duplicateUser);
    assertThrows(UsernameAlreadyExistsException.class, () -> {
        userService.saveUser(user, "ROLE_PUBLIC");
    });
    userRepository.delete(duplicateUser);
}
```

- **Test execution**:





2. Integration testing

Integration testing is conducted via the integration document where all the endpoints and rules of using the endpoints are listed. This serves as a contract between frontend and backend, here are some endpoint examples in the document.

PUT

api/books/edit/{bookId}

Note: The data in the payload will overwrite the old data in the DB. If the field is empty or not available in the payload RequestBody, it will overwrite the old data with null. Make sure you have all data in the PUT payload.

Book RequestBody Payload


```
{
  "book":{
      "userID": 2,
      "title": "test fixed",
      "isbn10": "1231231231",
      "isbn13": "1231231231231",
      "longDescription": "test",
      "shortDescription": "test",
      "publishedDate": "2021-01-01",
      "coverImageUrl": "https://something.com/abc.png",
      "status": "NEW",
      "price": 100.0,
```

```
      "quantity": 5
  },
  "authorNames":[
    "Reki Kawahara"
    ],
  "categoryNames":[
    "Romance"
    ],
```

POST

api/books/add

Book RequestBody Payload


```
[
  "book":{
      "userID": 2,
      "title": "test",
      "isbn10": "1231231231",
      "isbn13": "1231231231231",
      "longDescription": "test",
      "shortDescription": "test",
      "publishedDate": "2021-01-01",
      "coverImageUrl": "https://something.com/abc.png",
      "status": "NEW",
      "price": 100.0,
      "quantity": 5
  },
  "authorNames":[
    "Reki Kawahara"
    ],
  "categoryNames":[
    "Romance"
    ],
  "publisherName" : "Amazon"
```

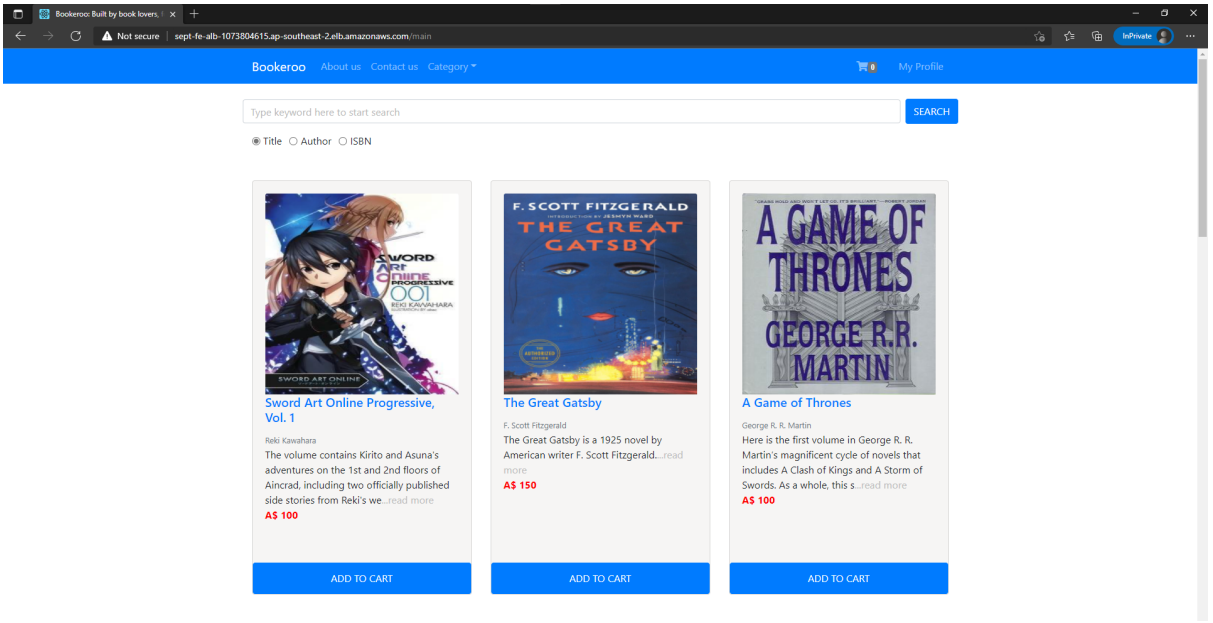

3. Browser testing

- **Browser**: Google Chrome and Microsoft Edge

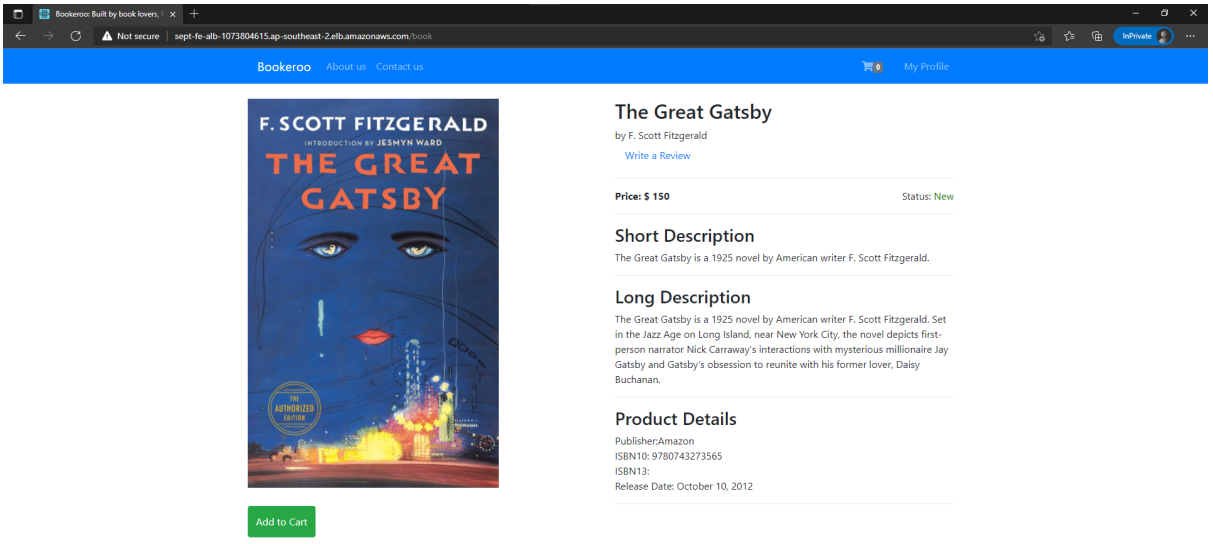
- **Result**: The application is rendered similarly on both browsers.

3.1 Microsoft Edge:

3.1.1 Rendering main page on Microsoft Edge

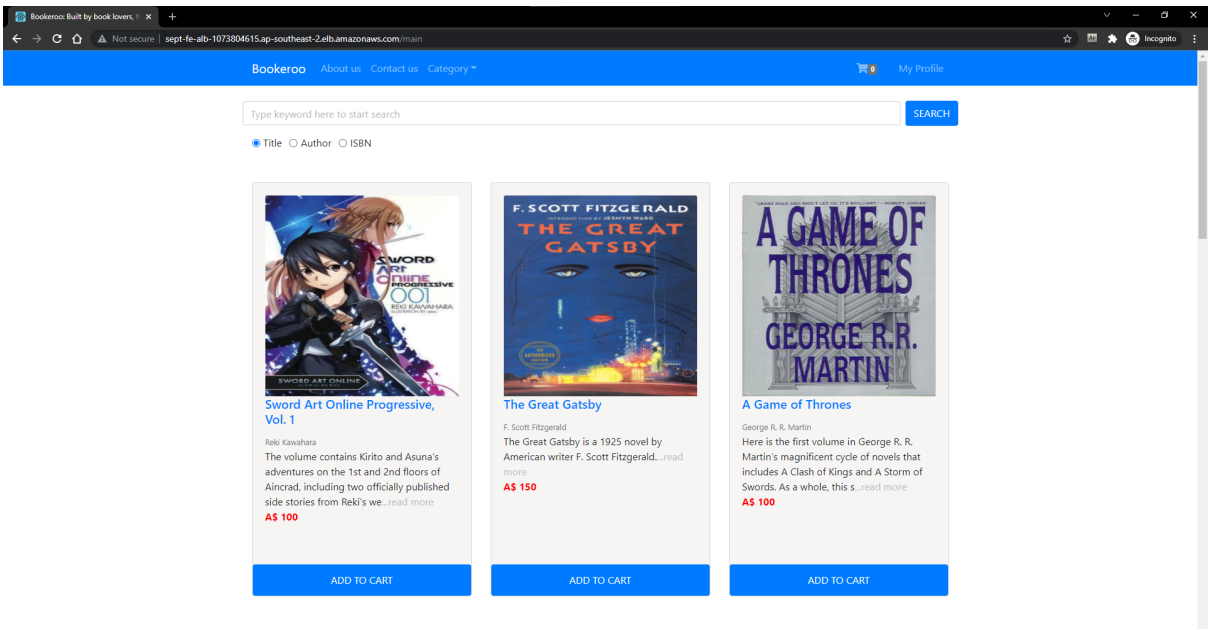


3.1.2 Rendering book page on Microsoft Edge

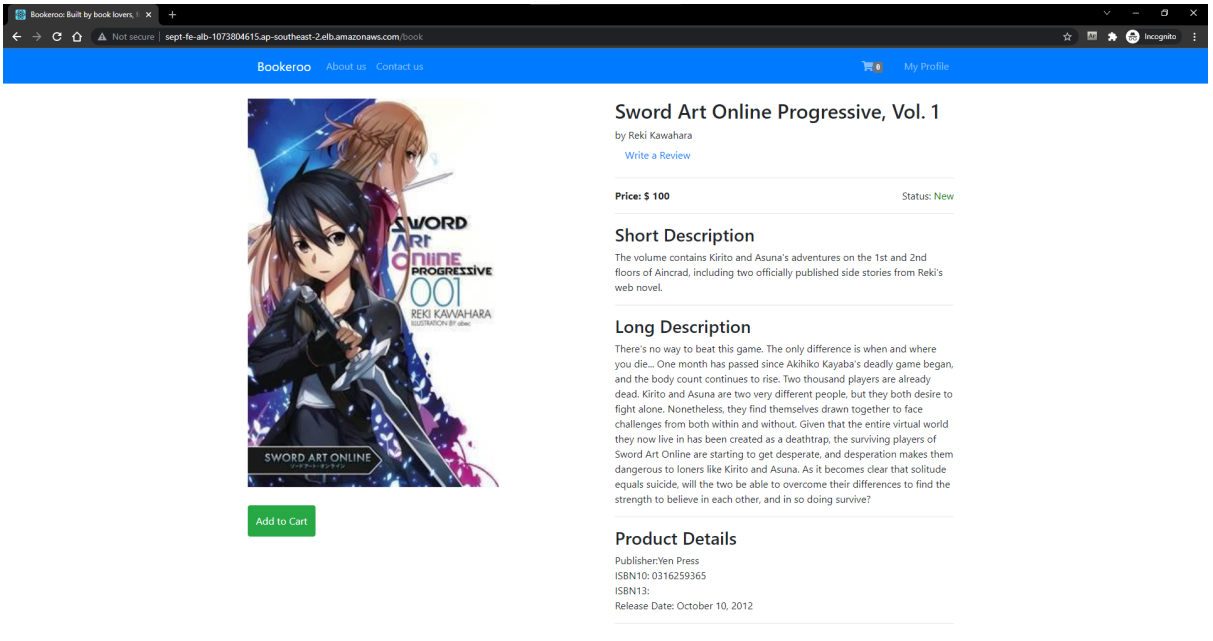


3.2 Google Chrome:

3.2.1 Rendering main page on Google Chrome



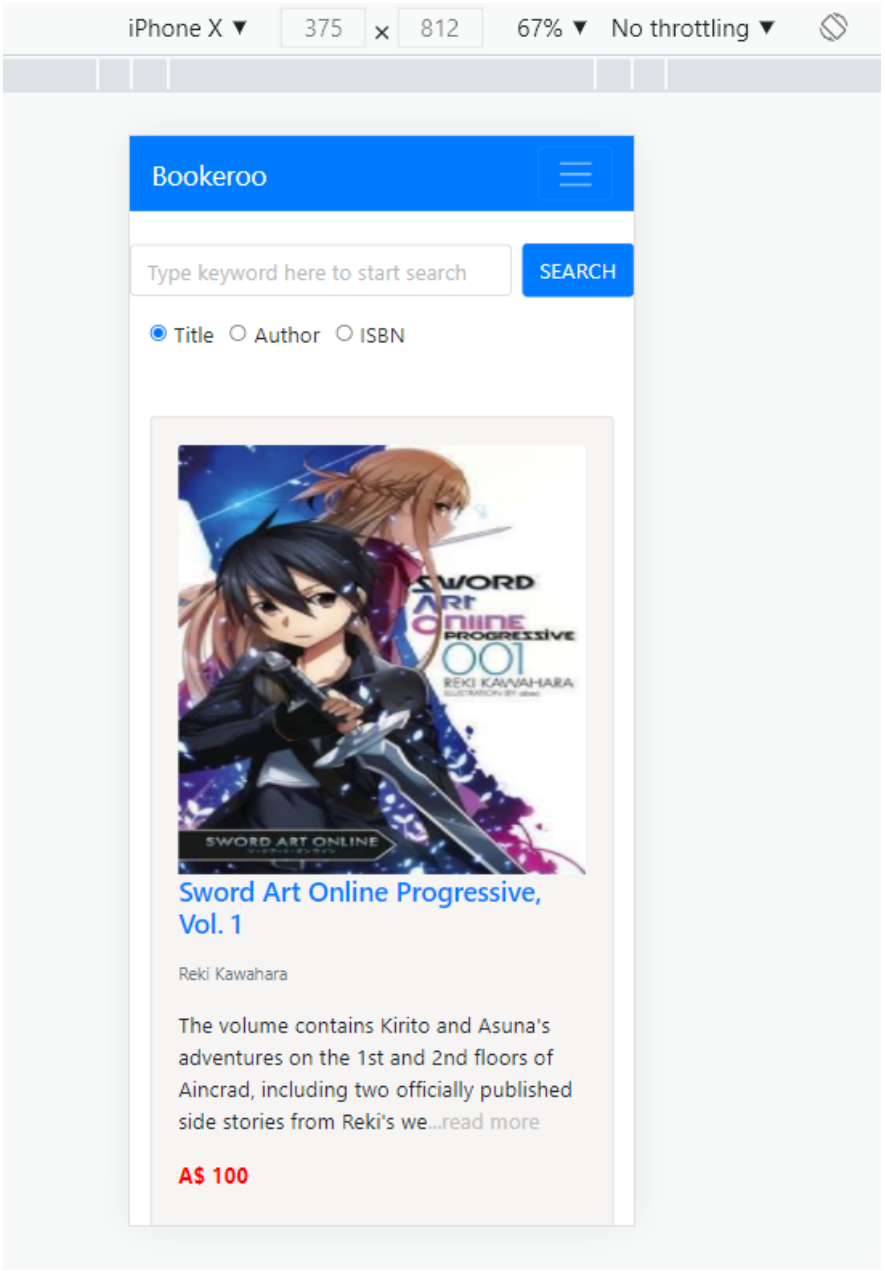
3.2.2 Rendering book page on Google Chrome



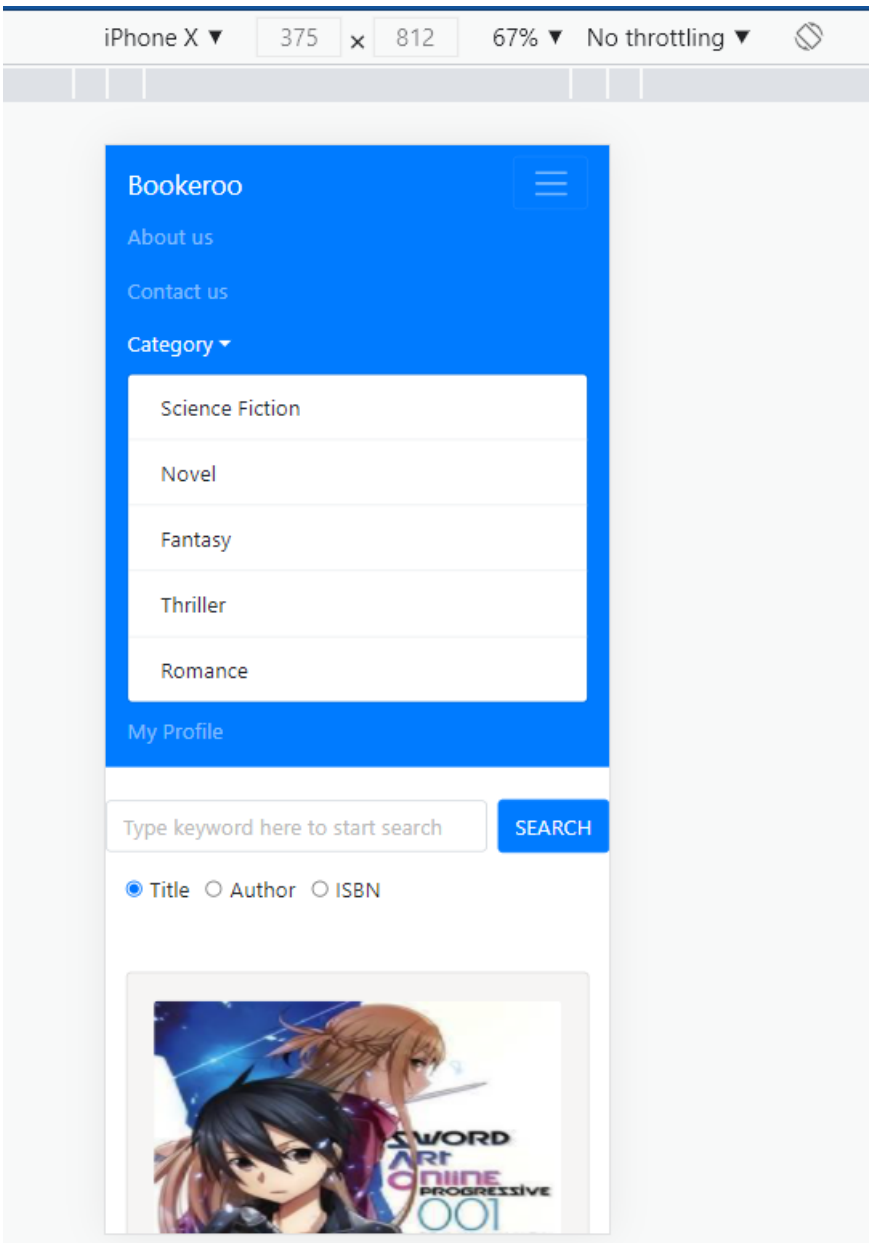
4. Mobile Device testing

- Tools: Google Developer Tools
- Device: IphoneX

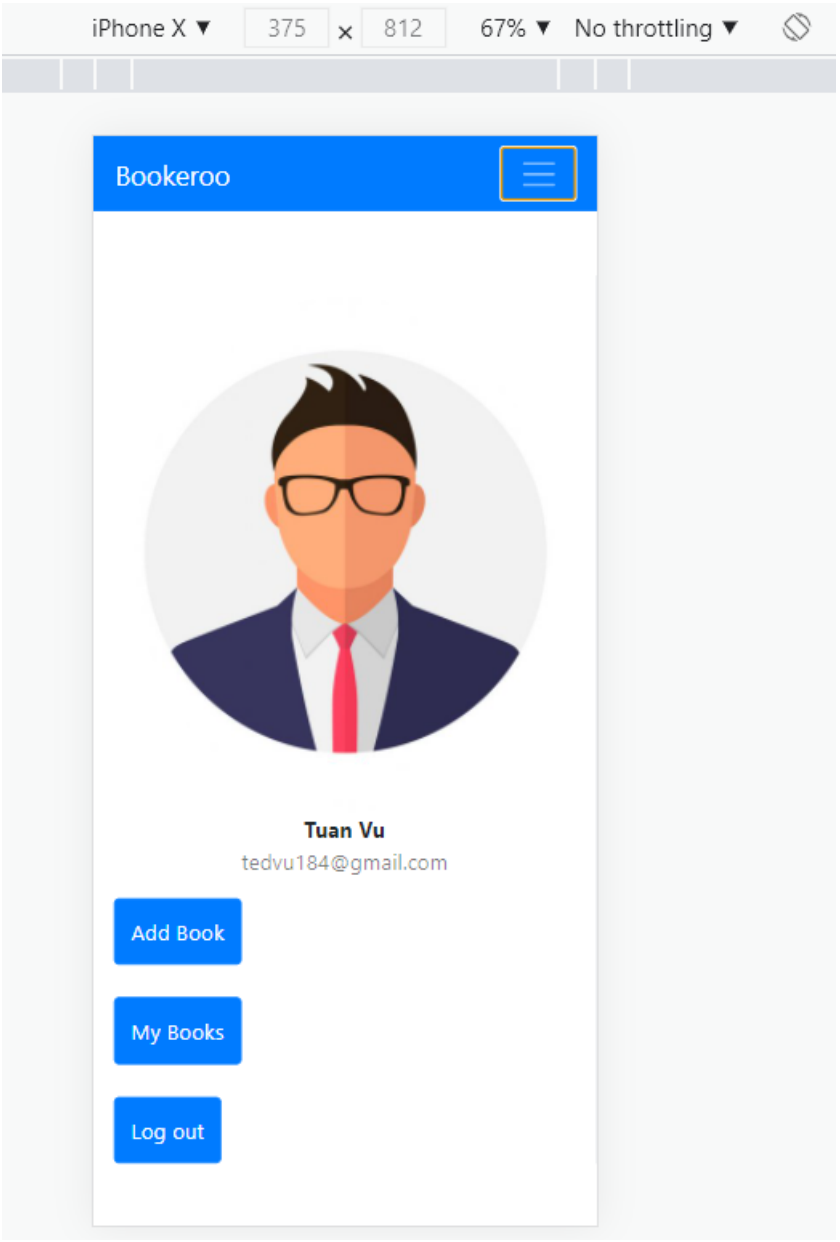
- Responsive rendering on IPhoneX



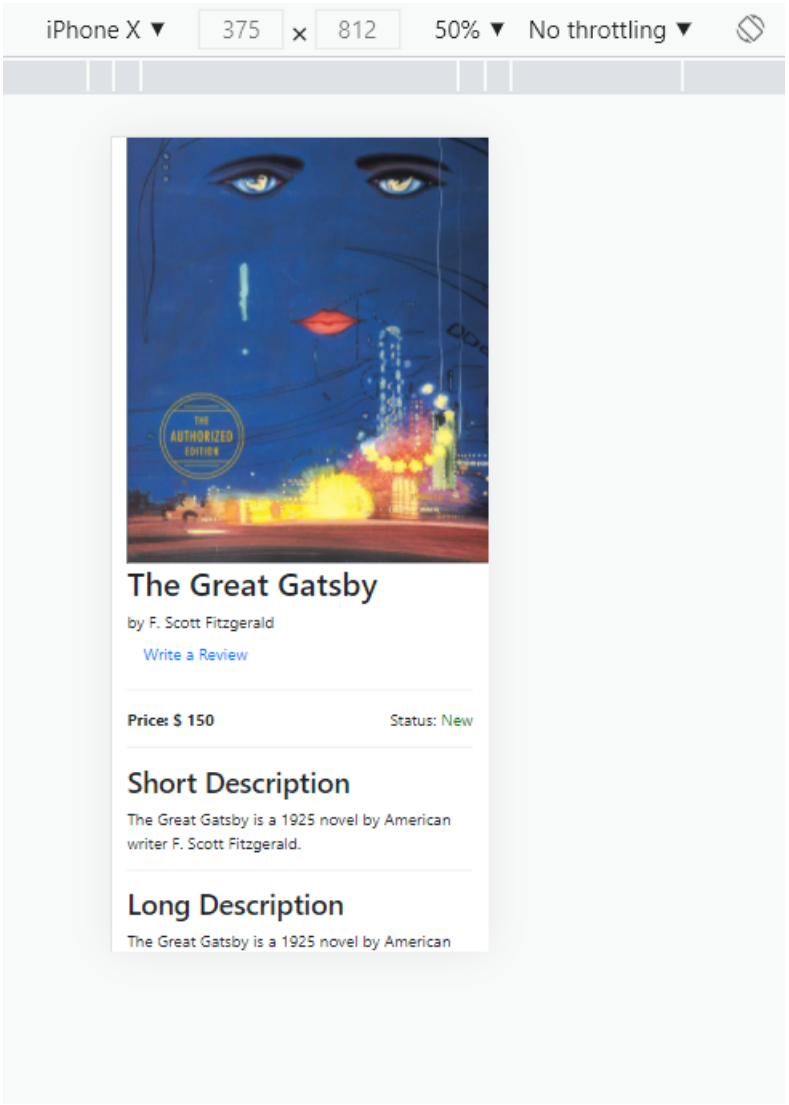
- The main page is rendering correctly, and all components are responsive on the device.



- The main page is rendering correctly, and all components are responsive on the device.



- The profile page is rendering correctly, and all components are responsive on the device.



- The book page is rendered correctly, and all components are responsive on the device.

5. Smoke testing

Here's a list of features that are included in the smoke testing

- Rendering shopping cart on the main page.
- Add a book to the shopping cart.
- Rendering Order history.
- Make a purchase of a shopping cart.

This feature will be tested before signing off with PO.

6. Acceptance testing

User story 7: As an Admin I want to be able to approve shop owners so that I can allow only legal users to be members.

Acceptance test:

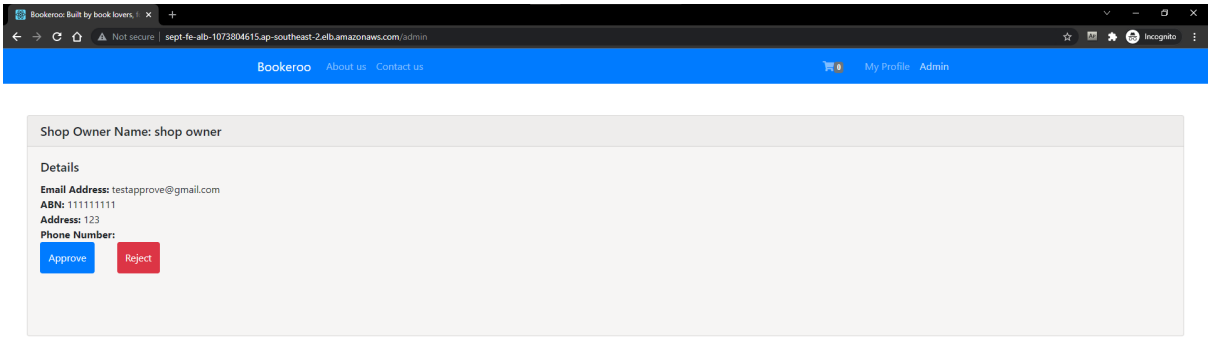
Given the admin has logged in and there is a pending shop owner on the admin page.

When the admin clicks on the approve button

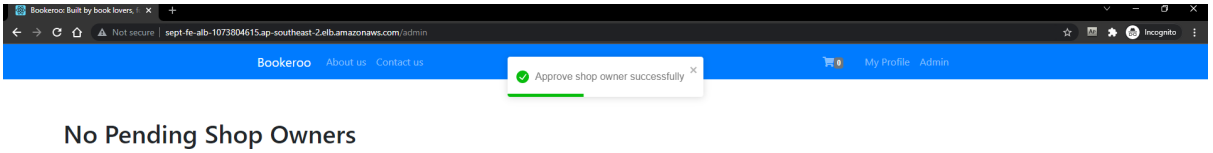
Then the user account is now approved and can log in with correct privilege as shop owner.

How does the implementation pass the acceptance test ?

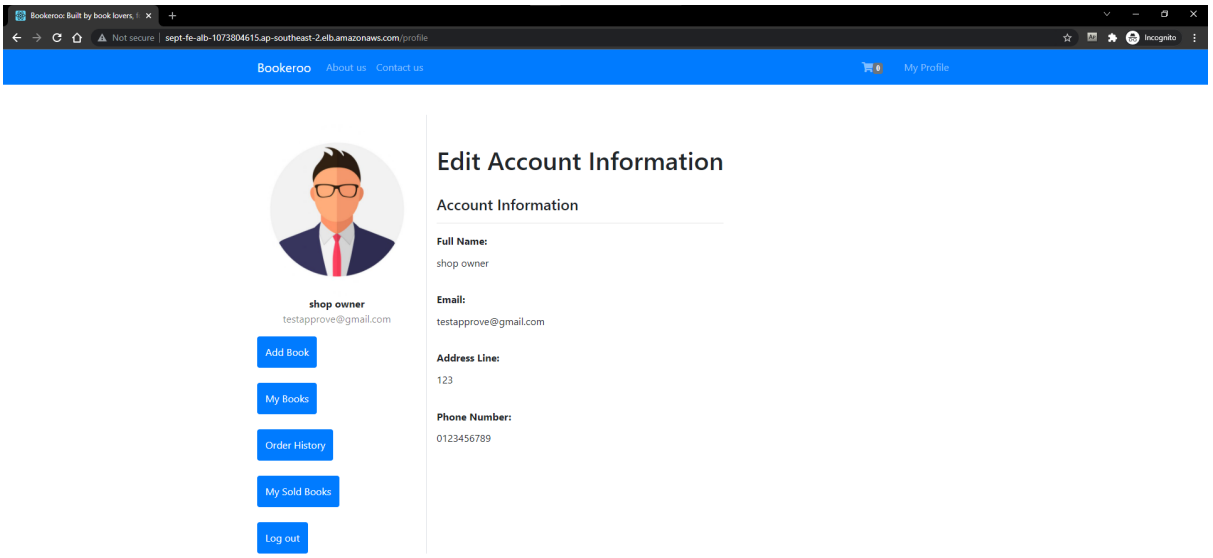
When the admin login and click on the admin page, he will see the pending shop owners:



Then the admin can click approve to approve this user.



The use can now log in:



User story 6: As a public user/shop owner, I want to see my transaction history so that I can keep track of the books I have purchased or sold.

Acceptance test:

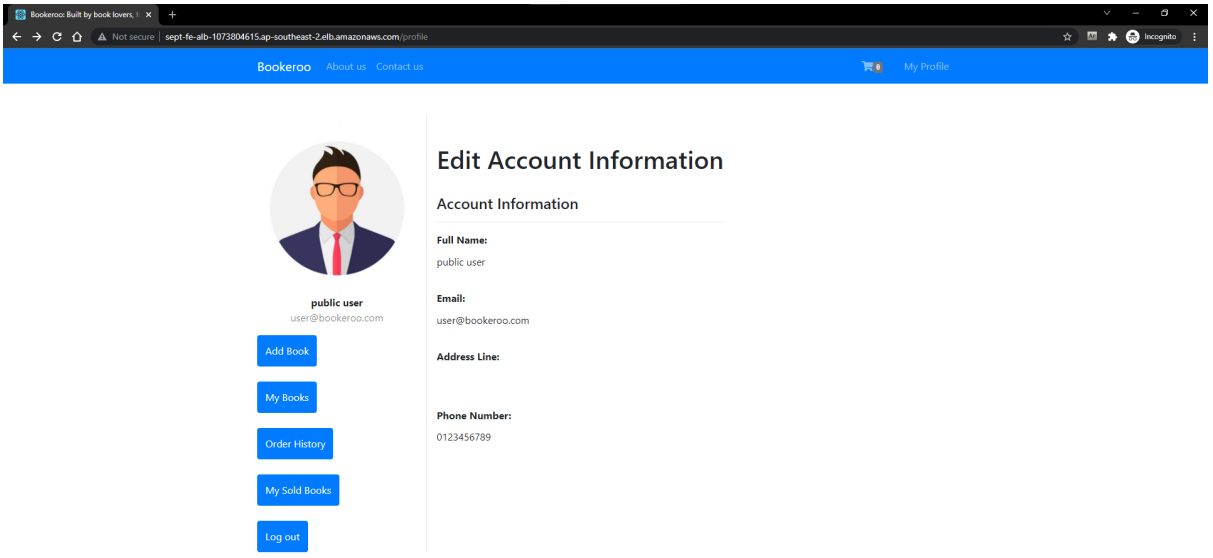
Given the public user/ shopowner has logged in and went to there profile page

When the admin clicks on the Order History/My Sold Books button

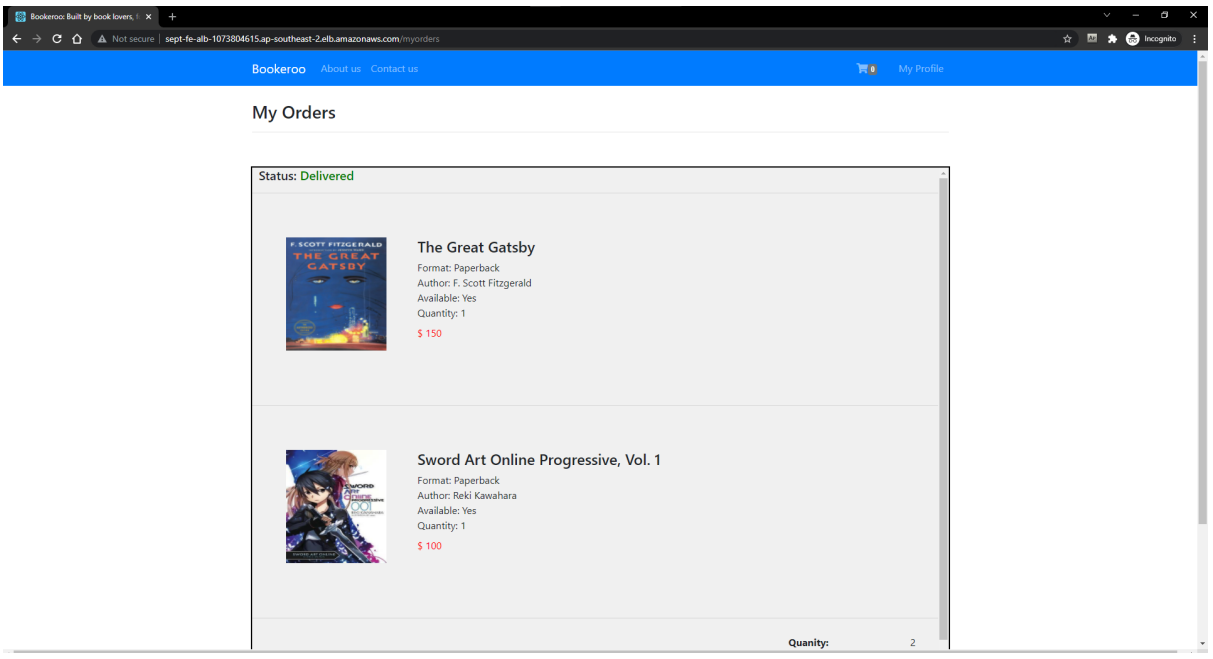
Then the user will see all the books they purchased or sold.

How does the implementation pass the acceptance test ?

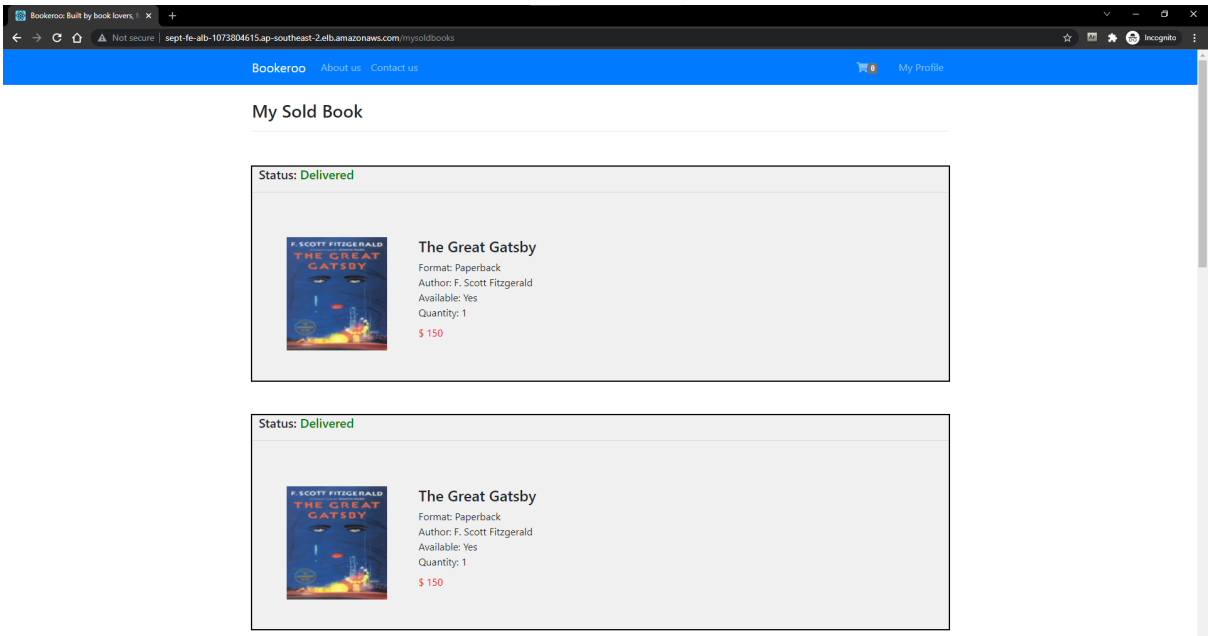
When the public user/ shopowner has logged in and went to there profile page



And they click the Order History button, they would see all their past orders:



When shop owners click on the My Sold Books button, they would see all their sold books.



User story 22: As a public user I want to add a book to my cart and buy book online so that I don't have to go to the bookstore to buy books.

Acceptance test:

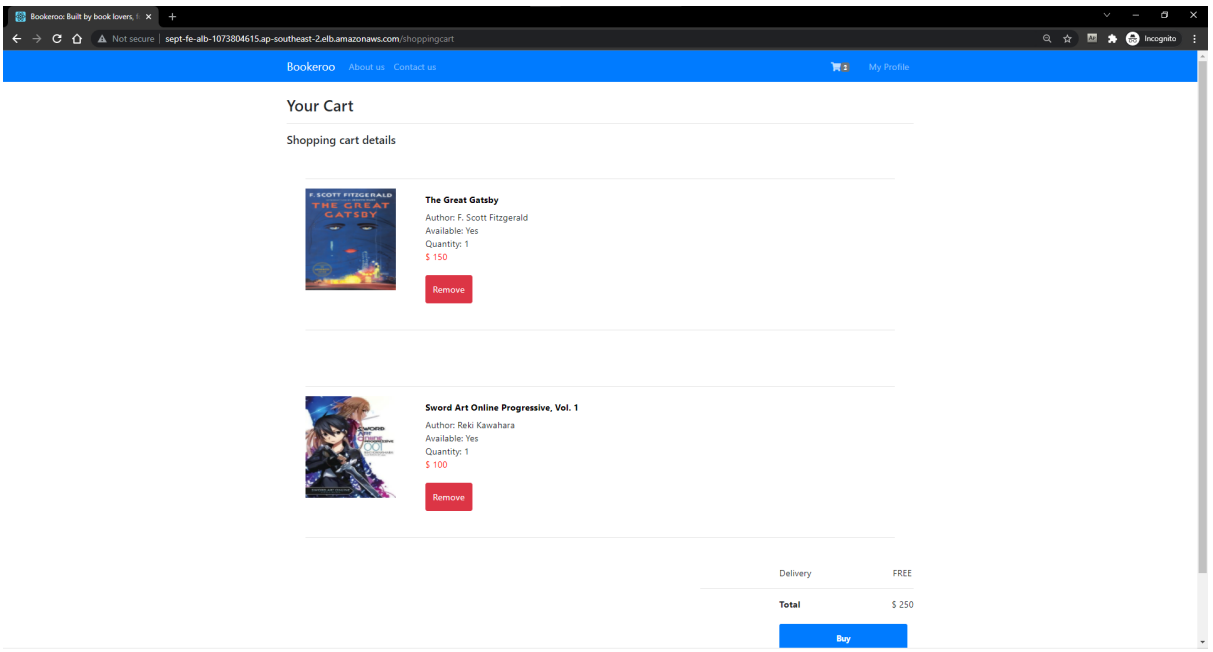
Given the public user has logged in

When the the users add books to cart and click on the pay button

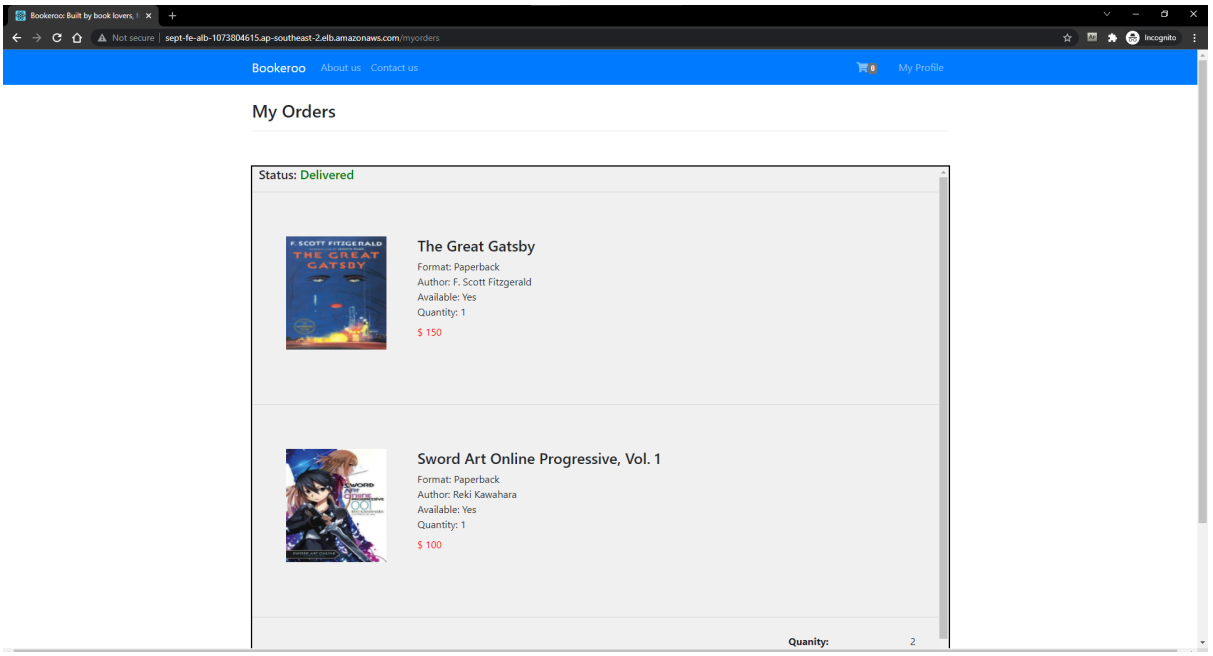
Then the transaction goes through and the user will see all the books they purchased.

How does the implementation pass the acceptance test ?

When the public user add books to there cart, they can see it in there cart



When the user clicks on the buy button, the transaction will go through, the order information would show up in both the buyer's order history and seller's sold books history.



Bookeroo: Built by book lovers, f

Not secure | sept-fe-alb-1073804615.ap-southeast-2.elb.amazonaws.com/mysoldbooks

Incognito

Bookeroo About us Contact us 0 My Profile

My Sold Book

Status: Delivered

The Great Gatsby

Format: Paperback
Author: F. Scott Fitzgerald
Available: Yes
Quantity: 1

$ 150

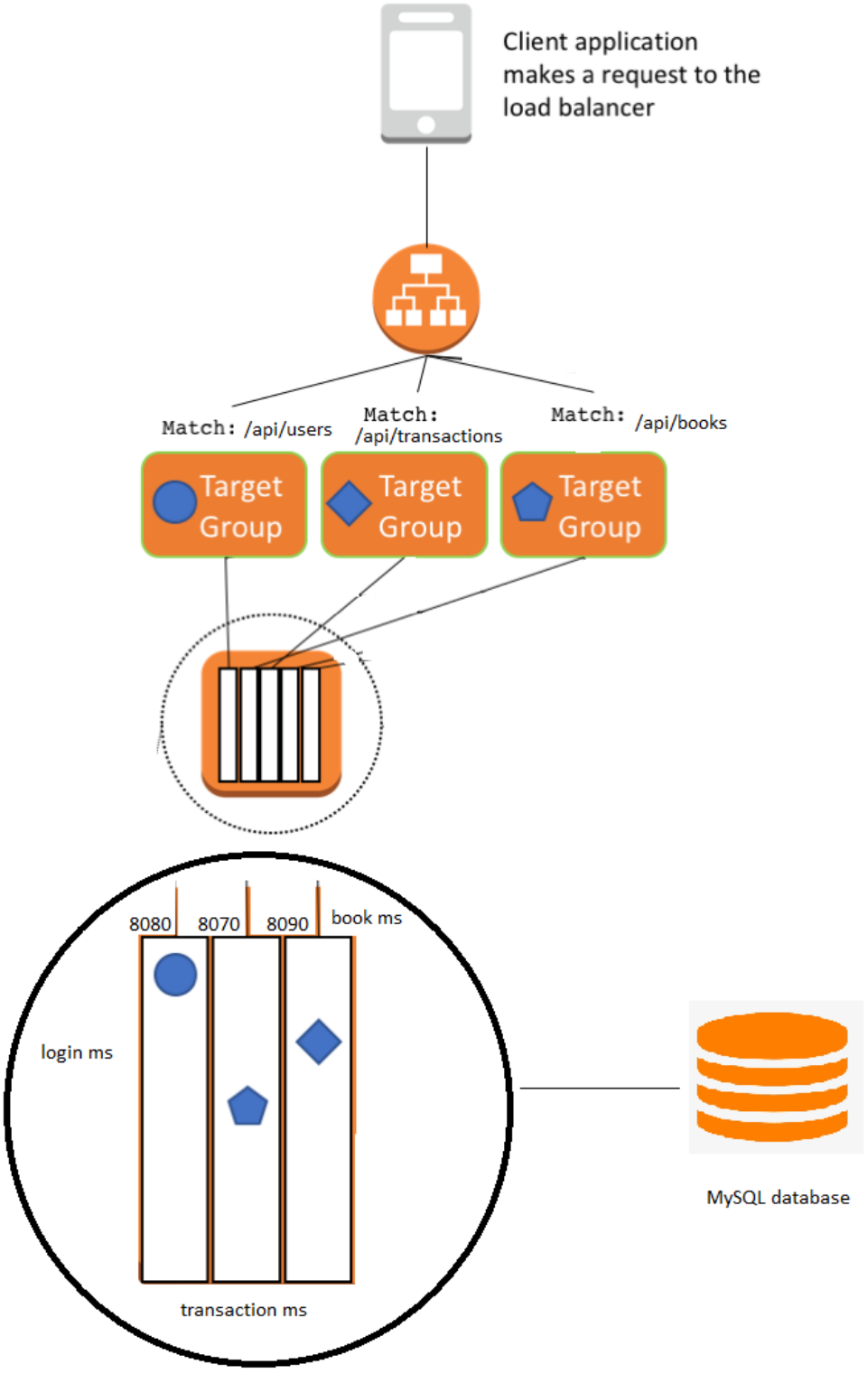
Status: Delivered

The Great Gatsby

Format: Paperback
Author: F. Scott Fitzgerald
Available: Yes
Quantity: 1

$ 150

SYSTEM ARCHITECTURE DIAGRAM



SCRUM PROCESS DESCRIPTION

- Our team meest at least twice a week, on Monday and Friday. More details on this could be referred to in the meeting minute part.

- Scrum master would be Ted for most of the time.

- A user story/feature would start in the product backlog in Jira, then moved to the sprint backlog, disassembled into smaller tasks and put in the To do column on Jira board. Then, as the team members worked on the tasks, it would be moved to the In Progress and Task done accordingly. Constant communication and updates would be made in the team scrum meeting, documented in the Meeting Minutes. After a feature is finished with CI/CD passed, the team would do integration and acceptance tests to make sure it complies with all the requirements. Then the pull request would be made and merged into the main code, marking a feature/user story done.

GITFLOW OVERVIEW AND ORGANIZATION

For the development of our program, we have used the gitflow model to organise our code and branches. We have two main branches in which we record the development of our program and merge our changes into, which are the master and develop branch. We start the development of a new feature by creating a new feature branch based on the develop branch, and merging when we finish. Aside from the feature branch, we also have other branches that we use; such as the bugfix, devops, and code review.

A quick summary of all the types of branches we have:

- Master: A full working copy of our program, where everything has been tested and works. Master branch stands at the top and is where we merge the develop branch into once we finish.
- Develop: The main development branch where we have the latest changes from the features which work. The develop branch is one step below the master branch, and is where we merge the changes from our feature branches.
- Feature: The branch where we push our new changes directly into, and requires a pull request to merge into develop. This is where developers directly work on.
- Bugfix: The branch where pushes our fix into, and merge into develop once we finish.
- Codereview: The branch where we do our code reviews and code optimisations, before we merge into develop.
- Devops: The branch where we make changes pertaining to the devops side of our development, such as pipeline (CI CD) changes, and docker related issues.

There is only one master and one develop, but multiple feature, bugfix, devops, and code review branches. We use the prefix feature/branch_name, bugfix/branch_name, devops/branch_name and codereview/branch_name as our naming convention.

Another naming convention that we have decided to adopt pertains to the naming of our development branches. When creating a branch, we also keep the format of:
- branchType/ticketNumber_taskDescription_assignee
- branchType/ticketNumber_assignee_taskDescription

In order to help with our branch organisation and make it easy to identify different branches. We also use a similar naming convention for our commits, where we always start the start of every commit message with the Jira ticket number which is assigned to the branch. This makes every commit message easy to identify and changes easy to keep track of.

DOCKER (BACKEND)

We have decided to dockerise our program to help our deployment. We have a dockerfile in each of our three microservices (loginmicroservices, bookmicroservice, transactionmicroservices), linked together by a docker-compose file in the parent folder of the backend src folder. Each of the dockerfile has an identical structure:

```
FROM maven:3.8.2-jdk-11
RUN mkdir /app

WORKDIR /app

COPY . .

RUN mvn -DskipTests=true package

ENTRYPOINT ["java","-jar","bookmicroservices/target/bookmicroservices-1.0.0-exec.jar"]
```

The docker first sets /app as the working directory, and copies everything in the root folder, before running a mvn package on the parent service dictated on the parent pom file. This is done because the microservices are dependent on each others' jar files to operate, and therefore require them to be available in the docker to run successfully. The parent pom files allow the jar files of all three microservices to be created and available for each microservice to use. This is also done by setting the build context of the service in the docker-compose file below and adding build dependencies in the services:

```yaml
version: '3'
services:
  login_service:
    build:
      context: ./
      dockerfile: ./loginmicroservices/Dockerfile
    image: 580827140556.dkr.ecr.ap-southeast-2.amazonaws.com/bookeroo-login-dev
    ports:
      - "8080:8080"
  book_service:
    build:
      context: ./
      dockerfile: ./bookmicroservices/Dockerfile
    depends_on:
      - login_service
    image: 580827140556.dkr.ecr.ap-southeast-2.amazonaws.com/bookeroo-book-dev
    ports:
      - "8090:8090"
  transaction_service:
    build:
      context: ./
      dockerfile: ./transactionmicroservices/Dockerfile
    depends_on:
      - login_service
      - book_service
    image: 580827140556.dkr.ecr.ap-southeast-2.amazonaws.com/bookeroo-transaction-dev
    ports:
      - "8070:8070"
```

In the docker, we have also set the name of the docker image in accordance with the image name generated in the AWS ECR, and also mapped the ports required for our APIs to function properly.

CI/CD (BACKEND)

We have two files/processes related to the CI/CD pipelines for our backend.

```yaml
# Workflow for Backend
name: CI - Backend (Build Maven package -> Run JUnit test, Build Docker Image)

on:
  push:
    branches:
      - master
  pull_request:
    branches:
      - Develop
      - master

jobs:
  build:
    runs-on:
      ubuntu-latest
      # Service containers to run with `runner-job`
    steps:
      - uses: actions/checkout@v2
      - name: Set up JDK 11
        uses: actions/setup-java@v2
        with:
          java-version: "11"
          distribution: "adopt"
          cache: maven
      - run: cd BackEnd/ && mvn install
      - name: Build LoginMicroservices and run JUnit tests
        run: cd BackEnd/loginmicroservices/ && chmod +x mvnw && ./mvnw package
      - name: Build BookMicroservices and run JUnit tests
        run: cd BackEnd/bookmicroservices/ && chmod +x mvnw && ./mvnw package
      - name: Build TransactionMicroservices and run JUnit tests
        run: cd BackEnd/transactionmicroservices/ && chmod +x mvnw && ./mvnw package
      - name: Build docker images
        id: build-images
        run: |
          # Build docker containers
          cd BackEnd/ && docker-compose build
```

The first file, CI-Backend, is mainly used for testing our codes by packaging the maven files, running unit tests, and building the docker images. This CI pipeline is run whenever a change is made in the master branch, or when a pull request is created towards the develop or master branch. This ensures everything works perfectly before merging and prevents broken codes from being merged into our main branches.

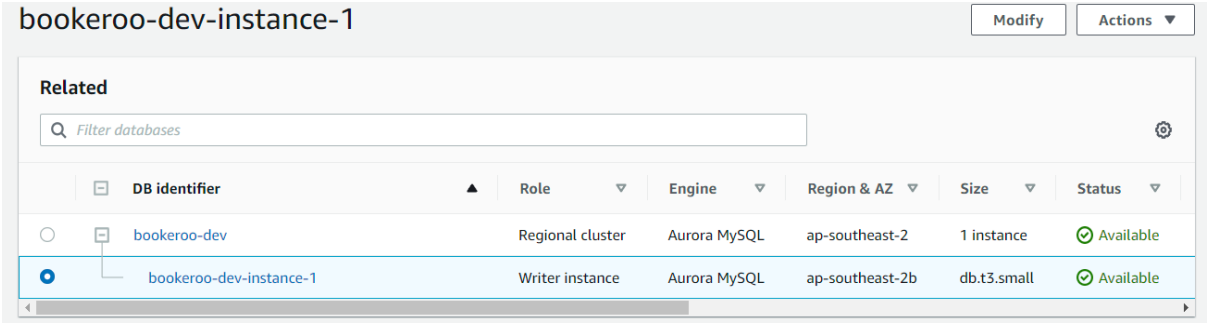
```yaml
# Workflow to build backend Docker image on CI server
name: CICD - Backend (Build Docker images and Push to ECR)

on:
  push:
    branches: [Develop]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout
        uses: actions/checkout@v2

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: ap-southeast-2

      - name: Login to Amazon ECR
        id: login-ecr
        uses: aws-actions/amazon-ecr-login@v1

      - name: Build and push the images to Amazon ECR
        id: build-deploy-images
        run: |
          # Build docker containers and push them to ECR
          cd BackEnd/ && docker-compose build
          docker push 580827140556.dkr.ecr.ap-southeast-2.amazonaws.com/bookeroo-login-dev
          docker push 580827140556.dkr.ecr.ap-southeast-2.amazonaws.com/bookeroo-book-dev
          docker push 580827140556.dkr.ecr.ap-southeast-2.amazonaws.com/bookeroo-transaction-dev
          # Force services to re-deploy using the new images
          aws ecs update-service --cluster ecs-bookeroo-dev --service bookeroo-service-login --force-new-deployment
          aws ecs update-service --cluster ecs-bookeroo-dev --service bookeroo-service-book --force-new-deployment
          aws ecs update-service --cluster ecs-bookeroo-dev --service bookeroo-service-transaction --force-new-deployment
```

The second pipeline is the CICD pipeline used for deploying our services into AWS. This pipeline is run whenever a change happens in the develop branch, mainly after a merge completes from a feature branch. The pipeline starts by configuring the AWS credentials using the keys saved in github, and logging into the account stated. The pipeline then proceeds to build corresponding docker images for the services using the docker-compose file, and push each docker image created into their own AWS ECR repositories. After everything is finished, the pipeline will tell each service in the AWS ECS cluster to force new deployment using the new images uploaded. This will force the service to deploy a new task with all the changes present from the new update.

DEPLOYMENT (BACKEND)

The deployment for the backend system can be largely divided into two parts:

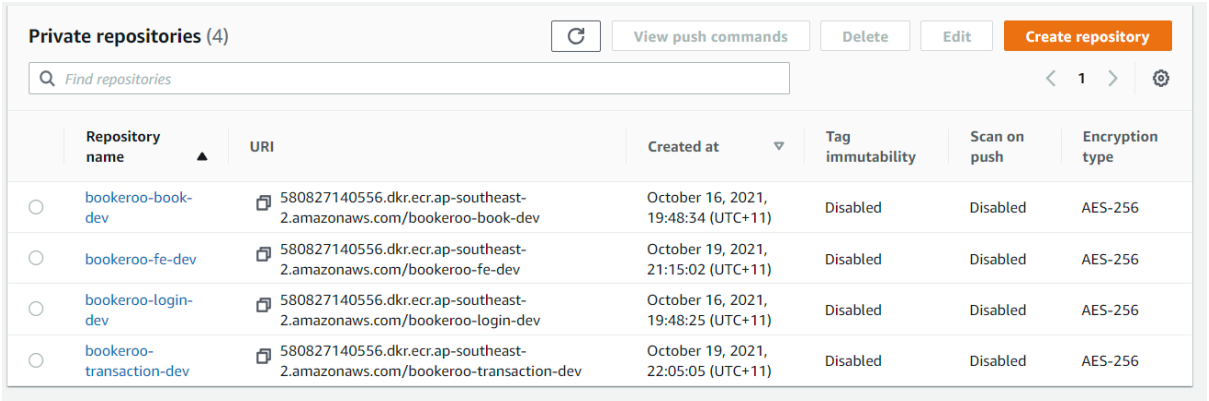
1. Database



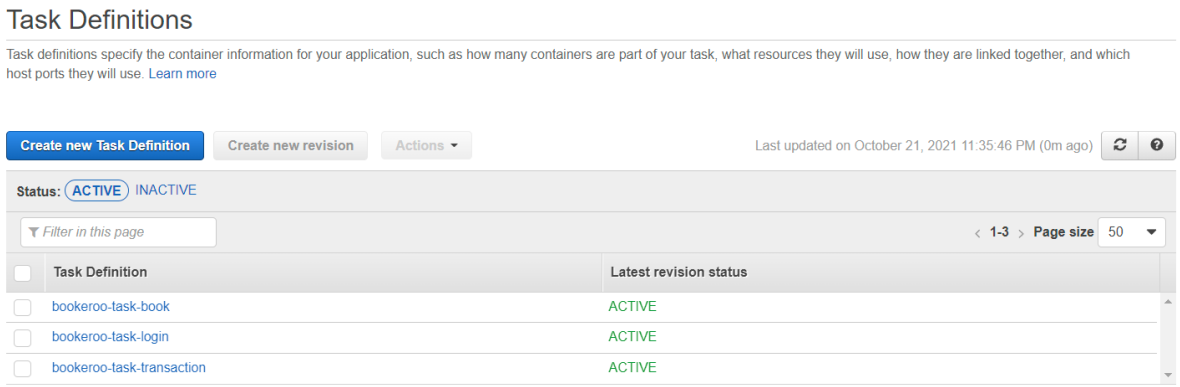
This is the simpler of the two and did not require much setup. We have used the Amazon RDS for our database, using the MySQL compatible Amazon Aurora as the engine, as per our previous development of using MySQL database for testing. The process was mostly straightforward, where we just replaced the jdbc connection in our application properties with the mariadb connector to the public DNS we were provided and everything worked perfectly.

2. Service

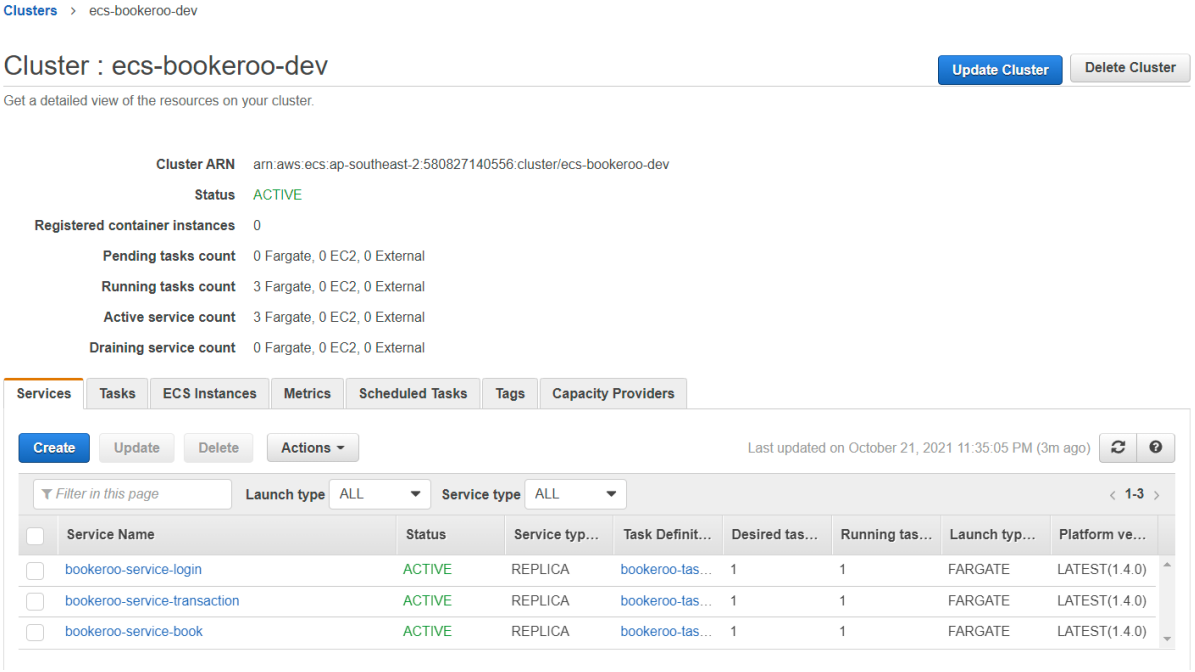
This is where things got a bit complicated. In order to deploy our services to AWS, there were several things that we had to set up. First was setting up the ECR.



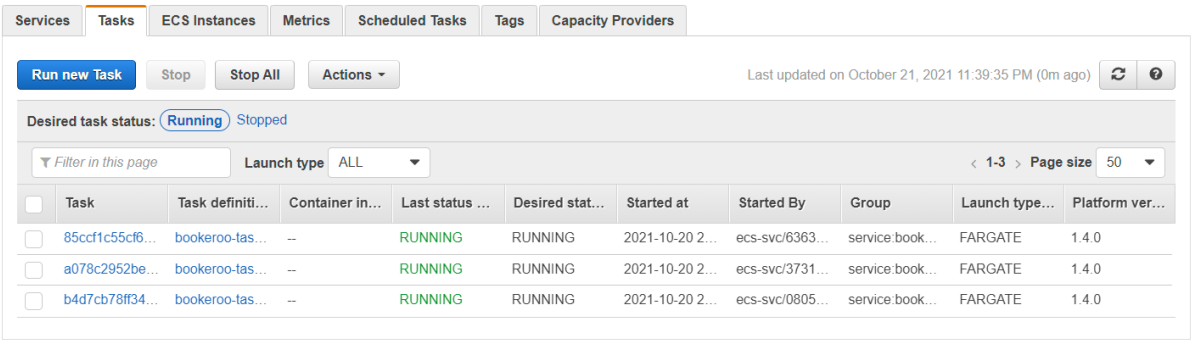
The ECR is basically an image repository for docker images, where we pushed the docker images for our services into.



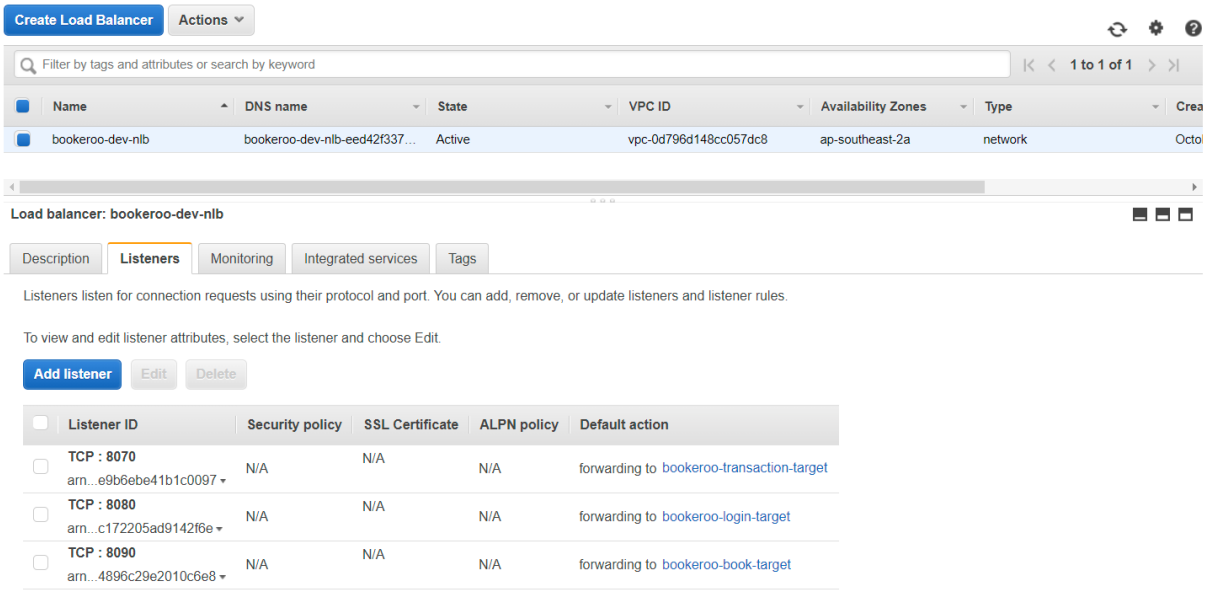
We then had to create a task definition for each of our services, where we define the resources required to run the task, also the ports and which docker image to run within the task. We listed the url for our ECR docker images in their corresponding task definitions in order for them to work correctly.

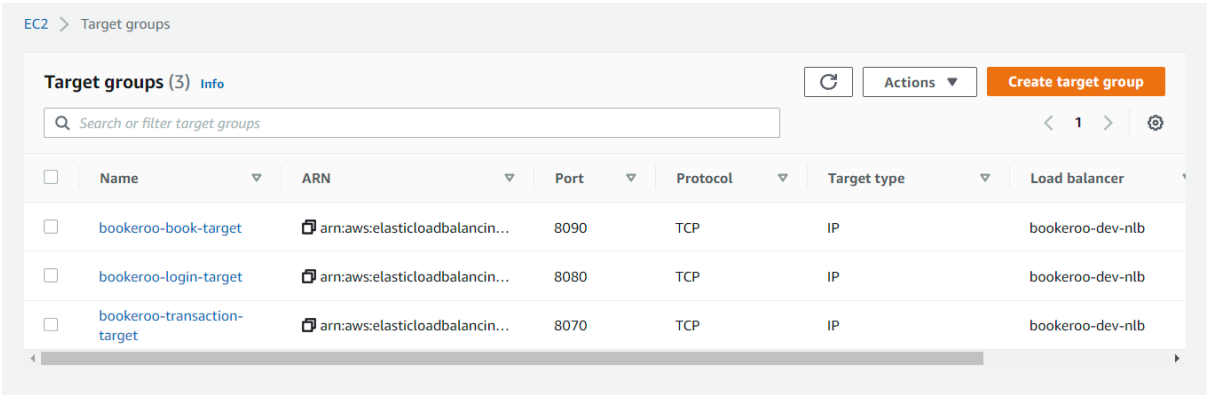


We then created a service for each of our task definitions, in order to automate the deployment for our tasks and support re-deployment on updates.

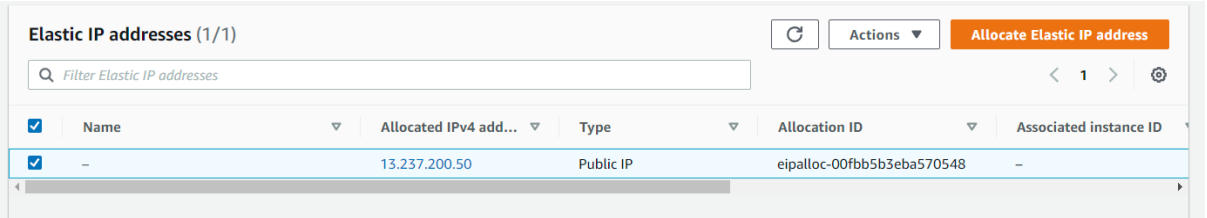


The service will automatically start the number of services we specify, in which we set each service to run one task each. However, after implementing this, we ran into a small problem, where the public DNS of our tasks keep changing after each re-deployment, causing issues with trying to integrate our APIs with the front-end. In the end, we have decided to fix this issue by integrating our services with a Network Load Balancer (NLB), and attaching an elastic IP address to it.



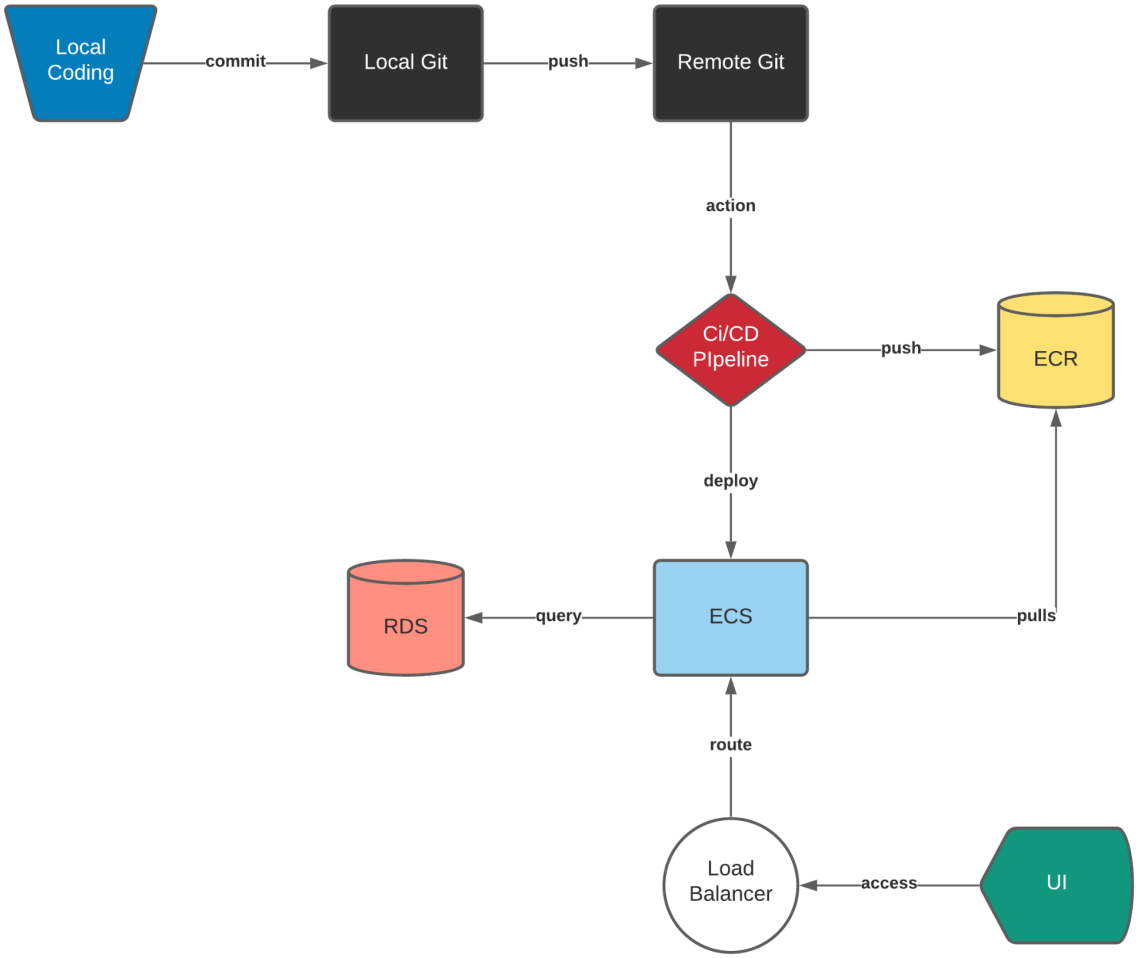


The load balancer requires the creation of target groups, and for them to be attached as listeners for the load balancer to properly forward requests on specific ports.



The use of elastic IP on the load balancer allows the re-deployed tasks to be accessed from the same public IP each time, which was essential for our deployment. With this, we have successfully deployed our services into AWS and tested everything to be fully working.

DEPLOYMENT DIAGRAM (BACKEND)



DOCKER (FRONTEND)

For docker frontend I used nginx as a proxy server to build the image, this enable faster frontend bundling.

```
FROM node:14-alpine AS builder
RUN mkdir /app
ENV NODE_ENV production
# Add a work directory
WORKDIR /app
# Cache and Install dependencies
COPY package*.json ./
RUN npm install
# Copy app files
COPY . .
# Build the app
RUN npm run build

# Bundle static assets with nginx
FROM nginx:1.21.0-alpine as production
# Copy built assets from builder
COPY --from=builder /app/build /usr/share/nginx/html
# Add your nginx.conf
COPY nginx.conf /etc/nginx/conf.d/default.conf
# Expose port
EXPOSE 80
# Start nginx
CMD ["nginx", "-g", "daemon off;"]
```

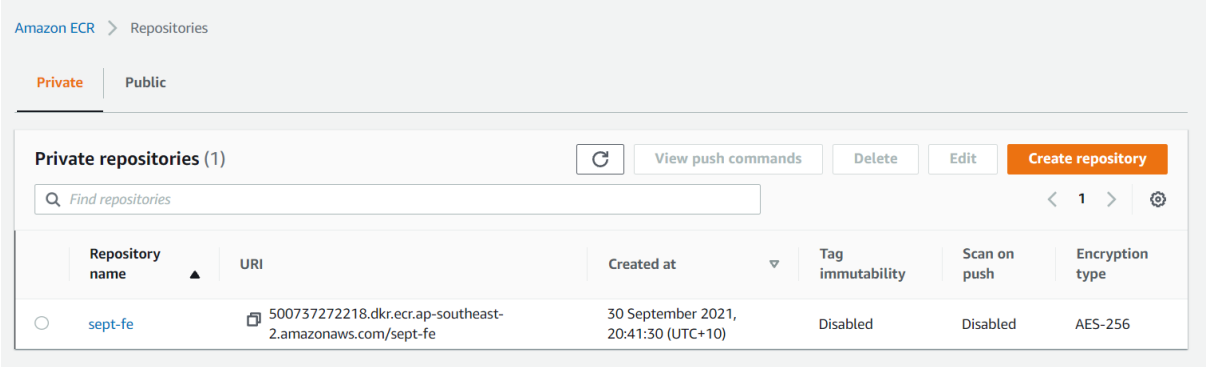
CI AND DEPLOYMENT (FRONTEND)

1. Service setup

The frontend CI/CD uses Docker for deployment, the CI will be triggered in every pull request into Develop and CD will be triggered in every merge into Develop. Here's the step of the pipeline:

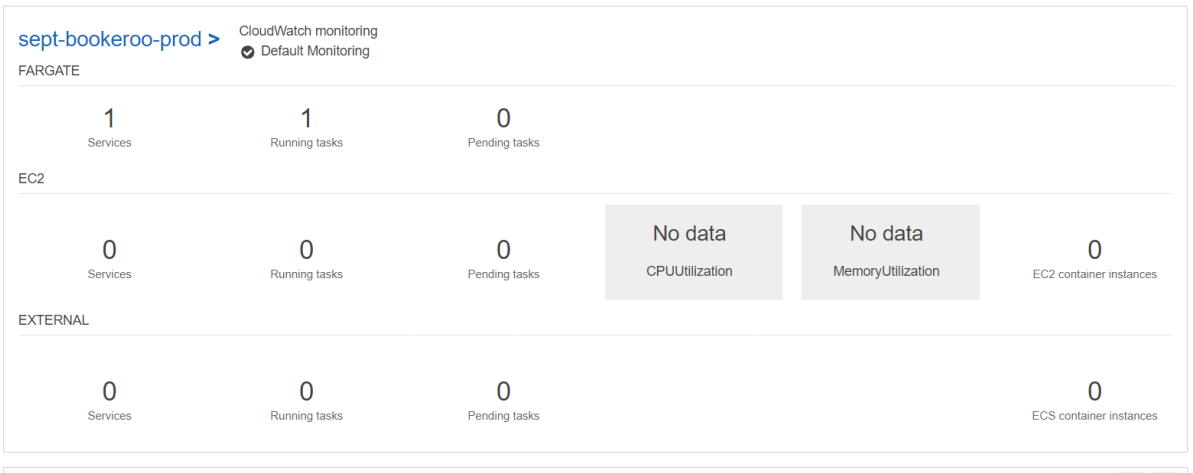
1.1 Push to ECR

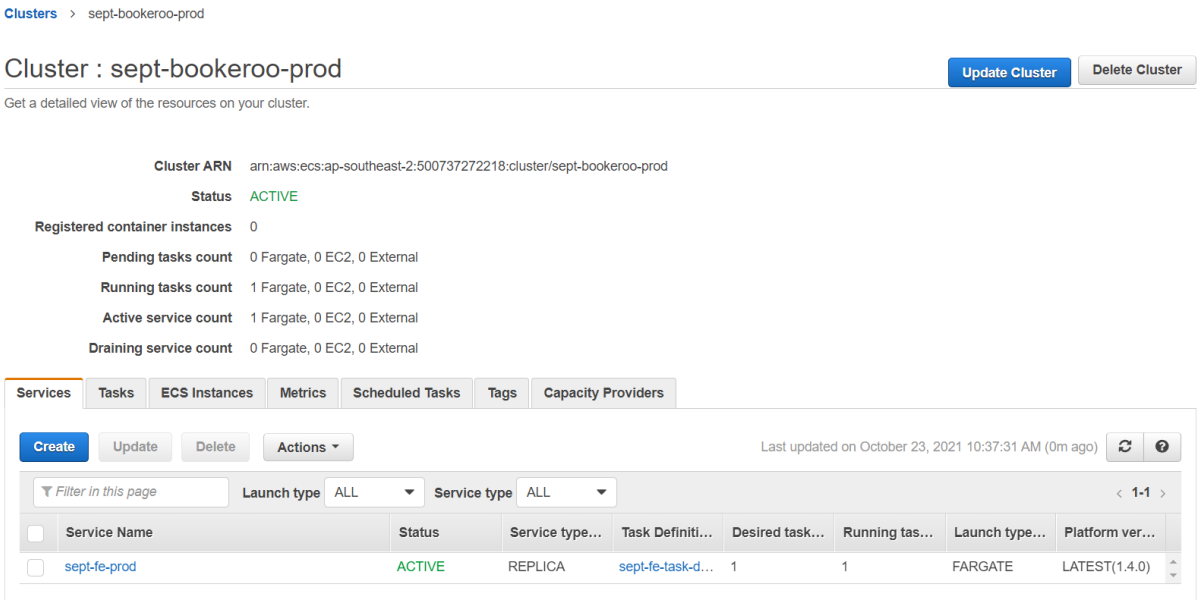
The latest image will be pushed to AWS ECR, AWS ECR is a service that hosts docker image provided from AWS.



1.2 Elastic Container Service deploys the image

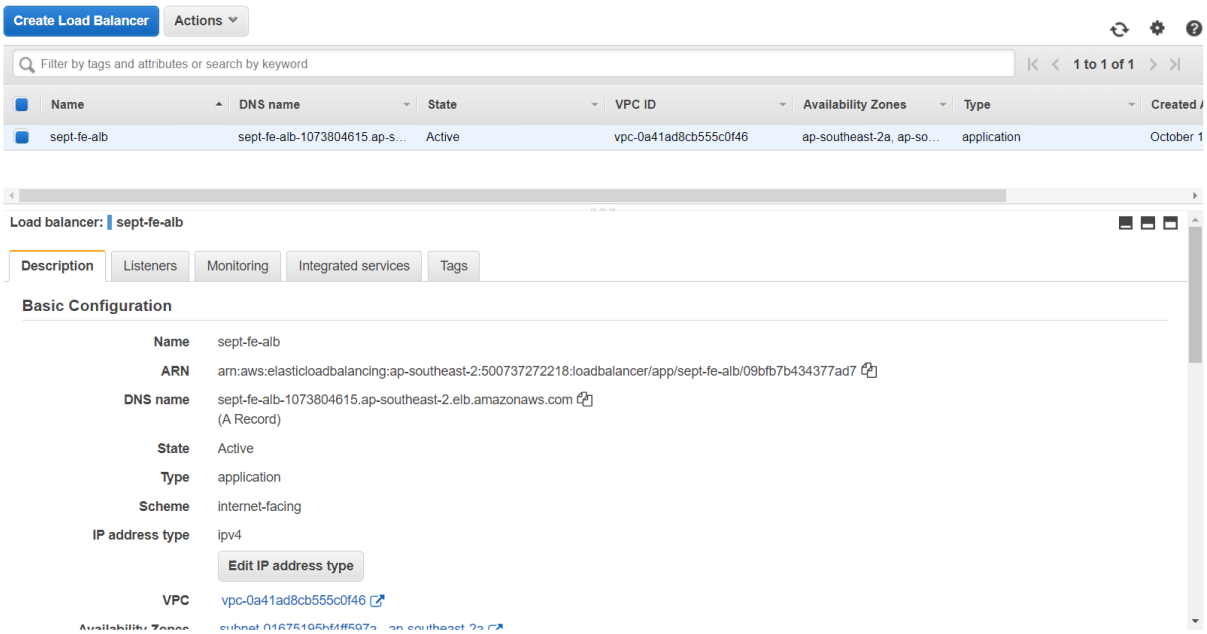
The ECS will pick up the latest image from ECR repo and deploy it with AWS fargate.





1.3 Load balancer provides public-facing access

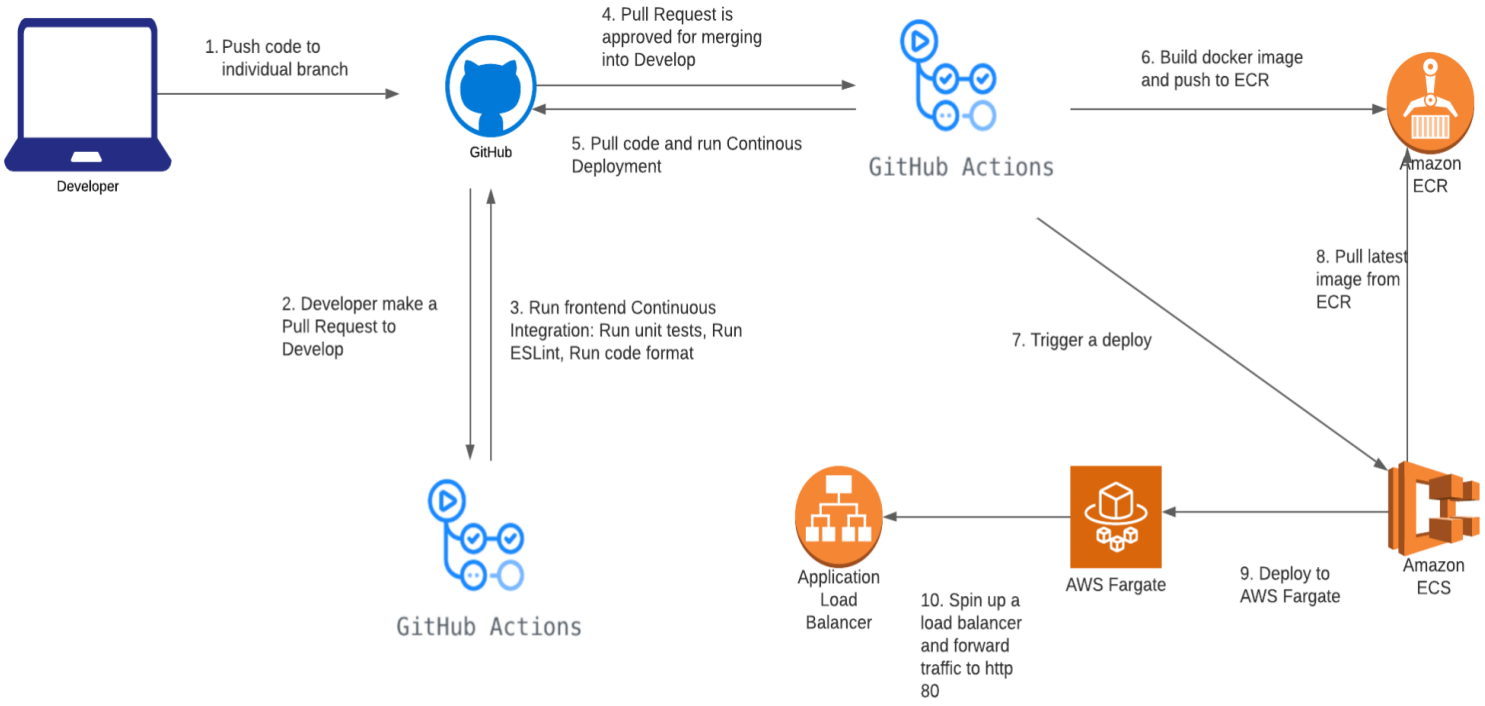
The frontend service is protected behind a Load Balancer with public access on port 80.



1.4 CI/CD file

```yaml
# Workflow to build and deploy Docker image
name: CICD - Frontend(Build Docker image and push to ECR then deploy the image)

on:
  push:
    branches: [Develop]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout
        uses: actions/checkout@v2

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID_FE }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY_FE }}
          aws-region: ap-southeast-2

      - name: Login to Amazon ECR
        id: login-ecr
        uses: aws-actions/amazon-ecr-login@v1

      - name: Build and push the images to Amazon ECR
        id: build-deploy-images
        run: |
          # Build docker containers and push them to ECR
          cd FrontEnd/myfirstapp/ && docker build . --file Dockerfile.prod --tag 500737272218.dkr.ecr.ap-southeast-2.amazonaws.
          docker push 500737272218.dkr.ecr.ap-southeast-2.amazonaws.com/sept-fe:latest
          aws ecs update-service --cluster sept-bookeroo-prod --service sept-fe-prod --force-new-deployment
```

DEPLOYMENT DIAGRAM (FRONTEND)



SOFTWARE ENHANCEMENT

Although we are satisfied with the software that we have produced over the 4-sprint period we have identified some aspects of the software that could have been improved:

- Provide more visual feedback for users particularly with the forms - we only use the generic input error message for any form error.
- Have more documentation regarding CI/CD, backend, frontend - we could have added a central knowledge repository for all team members to provide information on how CI/CD was set up, the API endpoints and repository structure of the frontend.
- Refactoring to a Design Pattern - we could have spent more time on refactoring to design patterns such as using MicroFrontend for Frontend.
- Creating two versions of Software - We plan to create two software versions one for Development the other for Production.
- Follow Deployment practice - We plan to follow AWS deployment practice and set up the CD to be more rigorous by applying Blue-Green Deployment practice.

PEER REVIEW

TEAM CONTRIBUTION

Team member	Contribution percentage	Contribution details
Tuan Vu	24	● Implementing frontend integration. ● Implementing frontend DevOps. ● Documentation.
Phu Nguyen	26	● Implementing the transaction microservice. ● Writing testing documentation: unit tests, acceptance test, browser test, ... ● Refactoring codes. ● Code review. ● Documentation
Janidu Higgoda	26	● Implementing static pages. ● Documentation. ● Testing frontend.
Edward Kahiro Kuo	24	● Implementing backend. ● Testing backend. ● Implementing backend devops. ● Documentation.