**Team 3 | Friday 10:30am | Tutor: Dipto**

|  |  |  |  |
| --- | --- | --- | --- |
| **Member** | **Student Number** | **Role** | **% Contribution** |
| Hibbaan Mohammed Nawaz | S3847243 | Front-End Developer | 25 |
| Ross Alexopoulos | S3839735 | Back-End Developer | 25 |
| Jeffrey Tan | S3851781 | Scrum Master, Front-End Developer | 25 |
| Allister Toos | S3843971 | Back-End Developer | 25 |

**Team Members + Peer Assessment**

**Links**

**Jira**

<https://bookeroo.atlassian.net/jira/software/projects/BOOK/boards/1>

**Teams**

<https://teams.microsoft.com/l/channel/19%3aQhGbNrgTOJSzYUYUzNyadTyARZ0x6OFaL9RpSzWaUm01%40thread.tacv2/General?groupId=1fa623d5-db2b-4891-b30c-4b3ecfb81383&tenantId=d1323671-cdbe-4417-b4d4-bdb24b51316b>

**GitHub**

<https://github.com/s3847243/Bookeroo>

**CircleCI**

<https://app.circleci.com/pipelines/github/s3847243/Bookeroo?invite=true>

**Items contributed**

|  |  |
| --- | --- |
| **Member** | **Items** |
| Hibbaan Mohammed Nawaz | Front-End and Tests for the following…  PBI: 4, 5, 6  Dashboards for Admins, Public Users, and Business accounts. Includes add/delete/edit for books and users. PBI: 18, 19 Contact-us with emailJS package.  About-us |
| Ross Alexopoulos | Back-End and Tests for the following…  PBI: 8, 9, 10 Books microservice with CRUD and search operations for books. Used in the dashboard and to view books in the main page. |
| Jeffrey Tan | Front-End and Tests for the following…  PBI: 1,2  Registration form for regular users or business users.  Login form.  PBI: 8, 10 Listing all the books in the catalogue.  Viewing a book’s details and listings made by users. PBI: 9, 22  Searching for specific books with a parameter.  Sorting books alphabetically or by year.  PBI: 24, 25  Adding a listing to the shopping cart.  Viewing the shopping cart. |
| Allister Toos | Back-End and Tests for the following…  PBI: 1, 2 Sign-up and Login via the users microservice.  PBI: 4, 5, 6  Users microservice to provide CRUD and approving accounts. Used in the admin dashboard. |

**CircleCI Screenshots**

Teams

Description automatically generated with medium confidenceGraphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application, email, Teams

Description automatically generated

**DockerHub**

Graphical user interface, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Summary of User Stories, Code, Tests, Functionality**

|  |  |
| --- | --- |
| **PBI** |  |
| **1** | As an unregistered user, I want to sign-up with my details, so I can create a public or business account with BOOKEROO. |
| Back-End | **Code:**  In ***loginmicroservices,***  UserController.java   * registerUser() handles web requests   userService.java   * holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_loginservices* |
| Front-End | **Code:**  Register.js   * React component to take form data and send API request to register user   actions/securityActions.js   * Holds the function to send API request   **Tests:**  Register.test.js |
| **2** | As a registered user, I want to enter my login information, so I can be directed to my application dashboard. |
| Back-End | **Code:**  In ***loginmicroservices,***  UserController.java   * authenticateUser() handles web requests   userService.java   * holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_loginservices* |
| Front-End | **Code:**  Login.js   * React component to take form data and send API request to login the user   actions/securityActions.js   * Holds the function to send API request and receive JWT token etc.   **Tests:**  Login.test.js |
| **4** | As an admin user, I want to approve/reject business accounts via my portal, so I can verify the legitimacy of a shop owner/publisher. |
| Back-End | **Code:**  In ***loginmicroservices,***  UserController.java   * getUnapproved(), approveUser(), blockUser() handles web requests   userService.java   * holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_loginservices* |
| Front-End | **Code:**  ApproveUsers.js   * React component to display unapproved users in a table and a button to approve them.   actions/dashboardActions.js   * getUnapprovedUsers()   **Tests:**  ApproveUsers.test.js |
| **5** | As an admin user, I want power to add/remove/edit users, so I can control account access on the platform. |
| Back-End | **Code:**  In ***loginmicroservices,***  UserController.java   * deleteUser(), allUsers(), updateUser(), registerUser() handles web requests   userService.java   * holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_loginservices* |
| Front-End | **Code:**  AllUsers.js   * React component to display users in a table with edit, block, delete functionality.   actions/dashboardActions.js   * getAllUsers(), deleteUser(), postEditUser()   **Tests:**  ApproveUsers.test.js |
| **6** | As an admin user, I want to add/remove/edit books, so I can control what information or listings are available to the public. |
| Back-End | **Code:**  In ***bookmicroservices,***  BookController.java   * createBook(), allBooks(), update(), delete() handles web requests   bookService.java   * holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_bookservices* |
| Front-End |  |
| **8** | As a customer, I want to see a book cover image preview with a table of contents, so that I can make an informed decision before buying an item. |
| Back-End | **Code:**  In ***bookmicroservices,***  BookController.java   * getByISBN() handles web requests   bookService.java   * getByISBN() holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_bookservices* |
| Front-End | **Code:**  BookDetails.js   * React component to render API data   actions/bookActions.js   * Holds the function to send API request with ISBN and return the book   **Tests:**  BookDetails.test.js |
| **9** | As public user, I want to search for books by name, author, ISBN and category, so I can view listings for books based on my interest. |
| Back-End | **Code:**  In ***bookmicroservices,***  BookController.java   * search() handles web requests   bookService.java   * search() holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_bookservices* |
| Front-End | **Code:**  BookSearch.js   * React component for the form   BookIndex.js   * Parent of the search-bar, sends the API request   actions/bookActions.js   * Holds the function to send API request with a query   **Tests:**  BookSearch.test.js |
| **10** | As any user, I want to see a list of all available books on the homepage, so I can browse the website's collection. |
| Back-End | **Code:**  In ***bookmicroservices,***  BookController.java   * search() handles web requests   bookService.java   * search() holds logic to interact with database   **Tests:**  In *test/java/com.rmit.sept.bk\_bookservices* |
| Front-End | **Code:**  BookIndex.js   * React component to render all books as links, with image previews.   actions/bookActions.js   * Holds the function to send API request to get all books   **Tests:**  BookIndex.test.js |
| **18** | As a public user, I want to contact the administrator so that I can contact about them and send a message to them |
| Back-End | N/A |
| Front-End | **Code:**  Contact.js   * React component to render a from which interacts with EmailJS |
| **19** | As a public user, I want to view the about page, to know more about the application and its owners |
| Back-End | N/A |
| Front-End | **Code:**  About.js   * React component to render our team’s information |
| **22** | As public user, I want to sort book lists by price + other parameters, so I can easily find the books I am looking for. |
| Back-End | N/A |
| Front-End | **Code:**  BookSort.js   * React component for the select elements   BookIndex.js   * Parent of the sort tab which sorts the books in the state   BookFunctions.js   * Functions used to shuffle and sort JavaScript arrays   **Tests:**  BookSearch.test.js |
| **24** | As a public user, I want to add a listing to my cart, so I can purchase the book later. |
| Back-End | N/A |
| Front-End | **Code:**  BookListings.js   * Child of BookDetails.js, renders the listings for that book as well as add to cart button   cartActions.js   * Functions to manage the cart in the browser’s localstorage.   **Tests:**  BookListings.test.js |
| **25** | As a public user, I want to view and edit my shopping cart, so I can preview the items I will to buy. |
| Back-End | N/A |
| Front-End | **Code:**  Cart.js   * React components to render the cart items, display price, and remove items.   cartActions.js   * Functions to manage the cart in the browser’s localstorage.   **Tests:**  Cart.test.js CartItem.test.js |

**Communication Logs**

Aside from weekly Stand-Up meetings (as demonstrated by the meeting minutes), our team conducted text-based communication via Discord (an app like Slack). To demonstrate our improvement in communication, the following are the log frequencies of last vs this sprint. The number of results is the number of messages sent in the text channel.

|  |  |  |
| --- | --- | --- |
| Member | Last Sprint | This Sprint |
| All members | Graphical user interface, text, application, chat or text message  Description automatically generated |  |
| Jeff |  |  |
| Mohammed |  |  |
| Ross |  |  |
| Allister |  |  |