**Books Shared Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Books Shared Application |
| Author | Divya Chadichal |
| Version | V1.0 |
| Creation Date | 09/15/2020 |
| Last Updated | 09/15/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc45989188)

[**1.** **Introduction** 3](#_Toc45989189)

[**2.** **Architecture Diagram** 3](#_Toc45989190)

[**3.** **Use-Case View** 4](#_Toc45989191)

[**4.** **Components** 4](#_Toc45989193)

[**5.** **Services** 5](#_Toc45989194)

[**6.** **Validations** 5](#_Toc45989195)

[**7.** **Tools and Libraries** 5](#_Toc45989196)

[**8.** **Unit Testing** 6](#_Toc45989197)

[**9.** **Responsive** 7](#_Toc45989198)

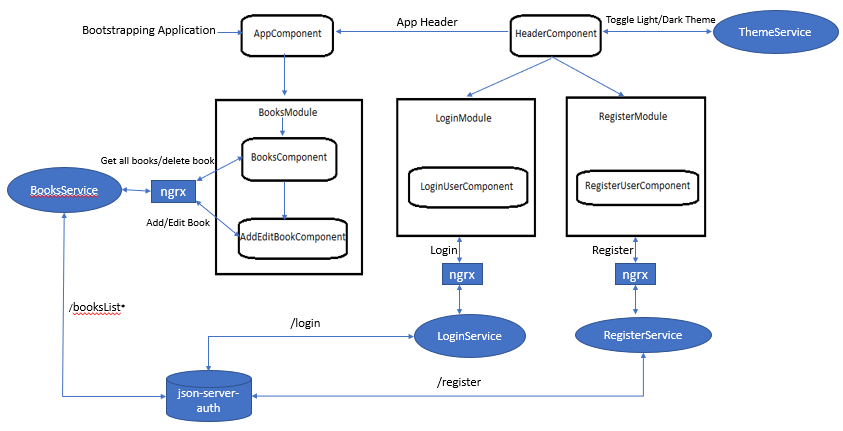
[**10.** **Accessibility** 7](#_Toc45989199)

[**11.** **Setup** 7](#_Toc45989201)

# **Introduction**

This document provides a comprehensive architectural overview of the Books Shared Application, using different architectural views to depict different aspects and functionalities of the application.

# **Architecture Diagram**



# **Use-Case View**

The application use cases are -

* **Login –** It provides functionality to Login to Books Shared Application.
* **Register –** It provides functionality to Register to Books Shared Application.
* **Display Books List** **–** Displays the List of books.
* **Add Book –** It provides functionality to add new Book details. Also ‘Add’ button is disabled until the required fields are valid.
* **Edit Book** **–** It provides functionality to edit/update the existing book details.
* **Delete Book** **–** Each Book listing will have a ‘Delete’ button, the respective Book can be deleted by clicking on it
* **Theme** – Can toggle between Light/Dark theme.

# **Components**

* **AppComponent** – This component is used for Application Bootstrapping
* **Header Component** – This component contains the application header, Login/ Register/ Theming buttons
* **BooksComponent** – Displays books list and contains buttons to add/edit/delete a book.
* **AddEditBookComponent** – This component contains form and logic to add/edit single book.
* **LoginComponent** – This component contains form and logic for Login functionality.
* **RegisterComponent** – This component contains form and logic for Register functionality.

Login/Register and CRUD operations on book uses ngrx for state management.

# **Services**

* **AuthGuardService** – Implements CanActivate for routing user to dashboard page if not logged in.
* **BookService** **–** This service has the logic on getting books list, add new Book details, edit/update Book details and delete entire book. All these functionalities call their respective json-server API’s.
* **LoginService** – This service has logic for calling json-server-auth Login API.
* **RegisterService** – This service has logic for calling json-server-auth Login API.
* **ThemeService** – This service has logic to toggle between Light/Dark theme.

# **Validations**

* Application has covered the required validation for Login/Register forms.
* Application shows proper error message for Login/Register failure i.e. incorrect email/password, duplicate email registration, short password etc.

**Validation strategy:**

* Reactive form for validation in Login/Register
* Display error message received from service
* Show Login/Register/Theming button if user is not logged in and show Logout/Theming if user is logged in.

# **Tools and Libraries**

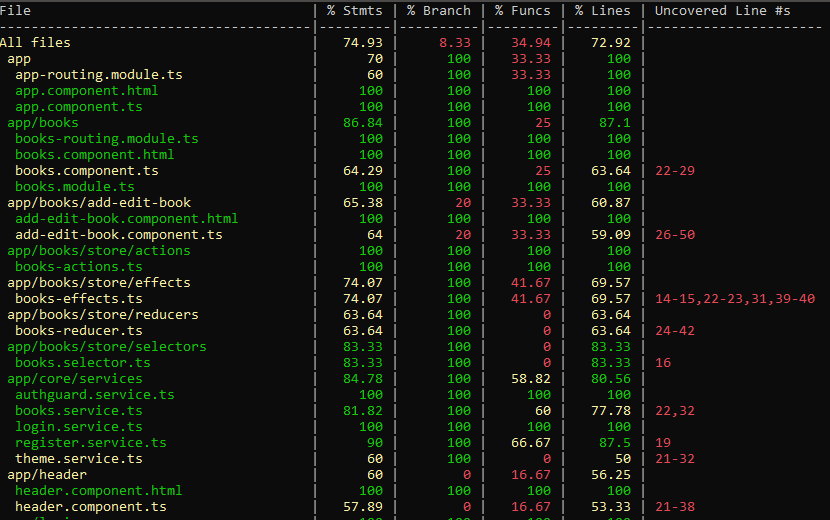
* Angular version - 10.1.0
* TypeScript - 4.0.2
* rxjs - 6.6.2
* ngrx/store – 10.0.0
* ngrx/entity – 10.0.0
* ngrx/effects – 10.0.0
* Jest – 26.4.2

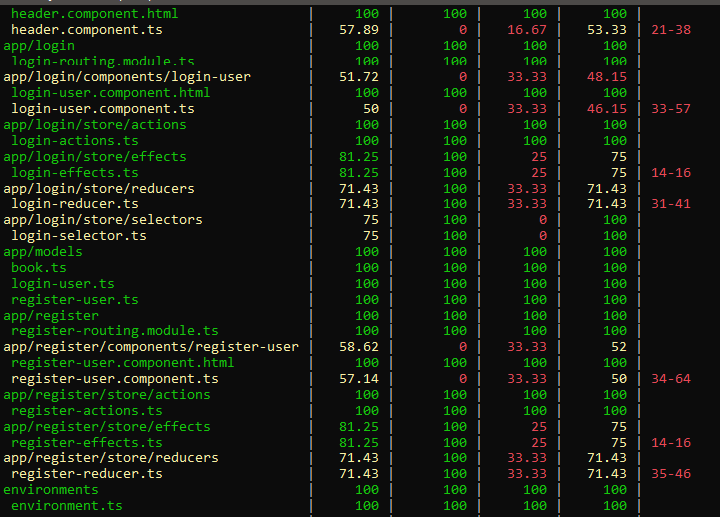
# **Unit Testing**

* Test cases for services: It includes test cases for the json-server API calls
* No of test cases: 11
* Component related test cases – 12
* Class related test cases – 3

**Total test cases: 26**

**Screenshots for unit test cases –**

****

****

# **Responsive**

* The application is responsive with all the device and desktop size.
* No Horizontal scrollbar for main application content.
* Used custom CSS for styling.

# **Accessibility**

For Accessibility, application is having appropriate HTML tags that are preferred for accessibility

User can run application via tab and enter.

# **Setup**

1. Download and install node.js (version: 12.18.3)
2. Install npm and install @angular (version: 10.1.0)
3. Git clone <https://github.com/arpitaPersistent/Sample-Application.git> origin master
4. Navigate to cloned folder & run **npm install**
5. Run **npm start**
6. For testing use **npm test**