**Sports Management Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Sports Management Application |
| Author | Vaibhav Bhandare |
| Angular Version | 10.0.7 |
| Creation Date | 09/7/2020 |
| Last Updated | 09/7/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc45989188)

[1. Introduction 3](#_Toc45989189)

[2. Architecture Diagram 3](#_Toc45989190)

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

[4. Flow Chart 5](#_Toc45989192)

[5. Components 6](#_Toc45989193)

[6. Services 6](#_Toc45989194)

[7. Validations 6](#_Toc45989195)

[8. Tools and Libraries 7](#_Toc45989196)

[9. Unit Testing 7](#_Toc45989197)

[10. Responsive 8](#_Toc45989198)

[11. Accessibility 8](#_Toc45989199)

[12. List other enhancement (at least top 3) 9](#_Toc45989200)

**13. Open Item…………………………………………………………………………………………………………………………….10**

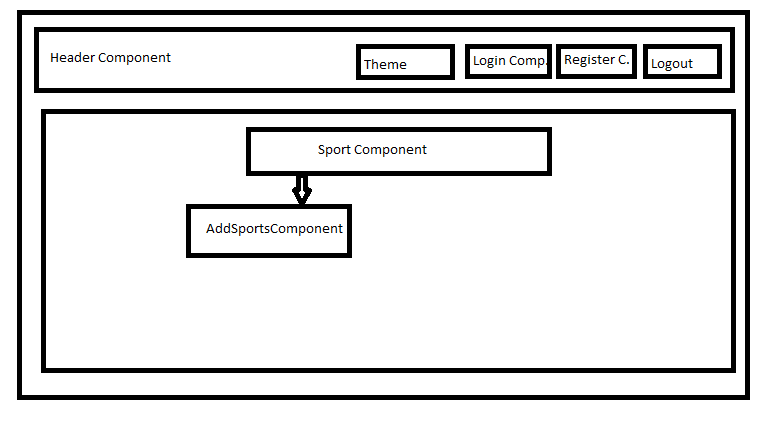
**14. Guards ………………………………………………………………………………………………………………………………….10**

15. Set Up……………………………………………………………………………………………………………………………………10

# **Introduction**

This document provides a comprehensive architectural overview and Sports Manage by User with the application, using number of different architectural views to depict different aspects of the application. You can perform multiple operations on the sports application.

# **Architecture Diagram**



# **Use-Case View**

The application use cases are -

* **Add Sport** – It provide functionality to add sport, also having unique name validation with it.
* **Edit Sport –** Any sport can be editable while clicking on sport button.
* **Delete sport** – Each sport will have a delete icon, the sport can be deleted by clicking on it, and refresh the sport table.
* **Toggle Theme** – The theme is change while clicking in toggle button, you can get dark and normal theme.
* **Log In –** These can login by user with there credential and open login form.
* **Sign Up –** These are register by new user while clicking it will open register form.

These use cases are initiated by the user.

**Features Uses**: -

|  |  |
| --- | --- |
| **Features Name** | **Uses/ Provide in Application** |
| NgRx | For Login, Sign up, CRUD |
| Jest for testing | For Testing Application |
| Accessibility | Use For all Application |
| Web components | Use for Component |
| Lit-Elements | Use for Component |
| Storybook | Installation done |
| Dynamic component loading | Use for Register Module |
| Json-server | Used for Application |

# **Flow Chart**

Sports List View with CRUD

Update Sport

|  |
| --- |
| Home Page |

Add Sport

# **Components**

* AppComponent – This component is used for Application root level.
* SportsComponent – Home Page of application to show all the sport and their respective details with edit and delete feature and adding new sport with all details.
* AddSportsComponent – This component contains two forms for add sport and update sport with submit and home button configuration.
* LoginComponent – This component contain form for login details with submit and home button.
* RegisterComponent - This component contain form for register details with submit and home button.
* HeaderComponent – This Component contain header title and button operations like login, toggle theme, sign up, logout configuration.

# **Services**

There are three services –

1. SportsListService – This service generally having logic about sports in the application and this contain code for getting only sport and sport with their table, edit sport details, add new sport and delete sport logic.

# **Validations**

1) Application covered the required validation for login, Register and Edit Form.

2) Application having validation for add Sport description i.e maxlength

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

- Required attribute true for individual form element

# **Tools and Libraries**

- Angular version: 10.0.6

- Node: 14.7.0

- TypesScript: 3.9

- ngrx/store: 10

- lit-element: 2.4.0

- Rxjs: 6.5.5

- jest: 26.4.2

- tslint: 6.1.0

- typescript: 3.9.5

# **Jest Testing**

Added Unit Test cases for services.

It consists one services -

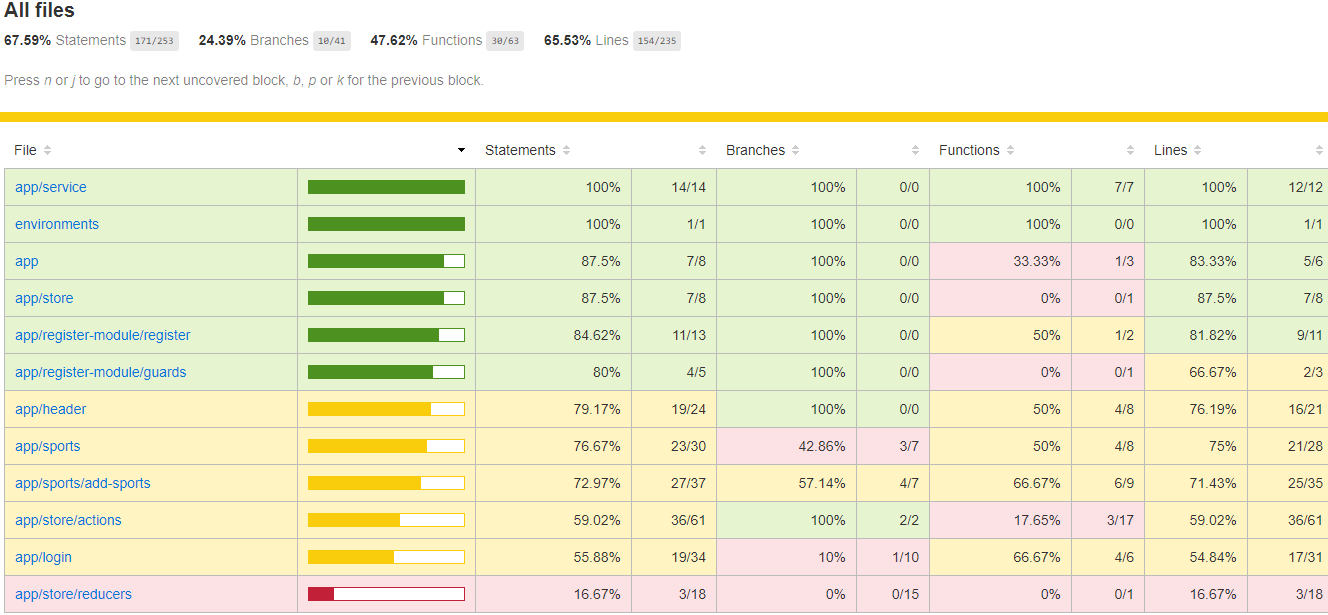
* + sports.service.ts – It contain the all logic like add sport, update sport, delete sport and get sport. And base URL is present. Test cases written in sports.service.spec.ts
    - No of test cases: 3

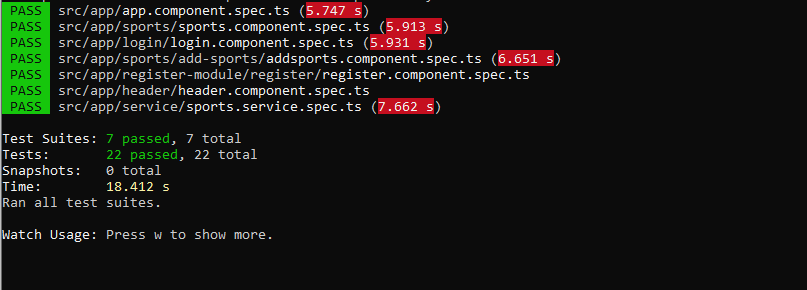
Added Unit Test cases for Component.

* + Each Component in Sport Application Consist their own test cases.

**Total test cases: 22**

**Screenshot for unit test cases -**





# **Responsive**

* The application is responsive with all the device and desktop size.
* No Horizontal scrollbar in application.
* Grid CSS Layout are used.

# **Accessibility**

For Accessibility, application having appropriate HTML tags like Button, input, table. etc this are preferred for accessibility.

User can run application via keyboard and tab.

# **List other enhancement (at least top 3)**

* **Feature enhancements** -

1. Only Admin Can access all Sports and Perform all CRUD operation as per

requirement.

1. Application design can enhance for better user experience.
2. Enhance application with user management and more accessibility.

* **Other enhancements** –

1. Server-side rendering support.
2. Client-side caching features can be enhanced
3. Service worker: [client-side caching and proxy, usually supports for offline capabilities].

**13. Open Item**

* i18

**14. Using Guards**

**- auth guard** – Is used for Register Component to confirm the navigation of route.

# **15. Setup**

1. Download and install node.js (version: 12.16)
2. Install npm and install @angular (version: 10.0)
3. Git clone <https://github.com/vaibhavbhandare/Assignment1-Sports> origin master
4. Navigate to cloned folder & run npm install
5. Run Json file i.e db.json on port 3000 used below json file location: -

* D:\Vaibhav\_Angular\Sports\src\assets\json
* And Run Command - json-server --watch db.json

1. Run ng serve
2. For Jest testing use Below Command: -

npm run test-watch - Show the all test cases Passed and Fail.