**Movies-CRUD Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Movies-CRUD Application |
| Author | Karthika karthikeyan |
| Version | V1.0 |
| Creation Date | 05Sep2020 |
| Last Updated | 07Sep2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc50366792)

[**1.** **Introduction** 3](#_Toc50366793)

[**2.** **Architecture Diagram** 3](#_Toc50366794)

[**3.** **Use-Case View** 3](#_Toc50366795)

[**4.** **Flow Chart** 5](#_Toc50366796)

[**5.** **Components** 6](#_Toc50366797)

[**6.** **Services** 6](#_Toc50366798)

[**7.** **Validations** 6](#_Toc50366799)

[**8.** **Tools and Libraries** 6](#_Toc50366800)

[**9.** **Unit Testing** 7](#_Toc50366801)

[**10.** **Responsive** 7](#_Toc50366802)

[**11.** **Accessibility** 7](#_Toc50366803)

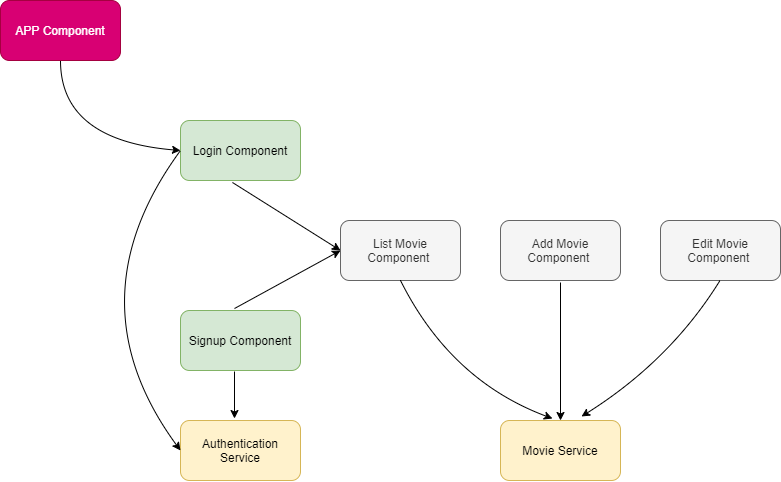
[**12.** **Theming** 7](#_Toc50366804)

[**13.** **Setup** 7](#_Toc50366805)

# **Introduction**

This document provides a comprehensive architectural overview of the application, using number of different architectural views to depict different aspects of the application.

# **Architecture Diagram**



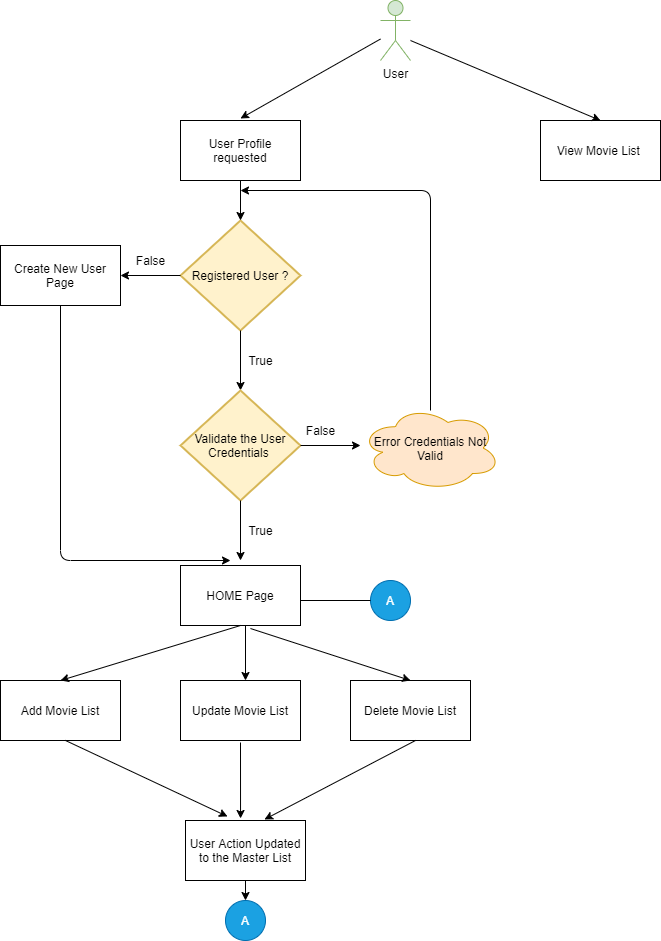
# **Use-Case View**

The application use cases are –

* **Login** – It allows user to logged in to perform the actions like to perform crud operations
* **Registration** – If user is not existing, We can register the user to perform crud operation
* **Add Movies** – It provide functionality to add movie.
* **Edit Movie –** Any movie can be editable while clicking on edit button.
* **Delete Movie** – Each movie will have a delete icon, the movie can be deleted by clicking on delete button

These use cases are initiated by the user.

# **Flow Chart**



# **Components**

* AppComponent – This component is used for Application Bootstrapping
* ListMoviesComponent – Home Page of application that show all the list and their respective movies with edit and delete feature and links for adding new movies.
* AddMovieComponent – This component contain form for add movie.
* EditMovieComponent – This component contain form for edit movie.

# **Services**

There are three services –

1. MovieService – This service generally having logic about movie in the application and this contain code for getting movie list, edit movie details, add new movie and delete movie logic.
2. AuthService – This service generally having logic about login into the application and this contain code for Registration logic.
3. JsonServer – Used db.json for performing all the tasks

# **Validations**

1) Application covered the required validation for login and movie.

2) Application having validation for login user whether they have the login details or not

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

- Required attribute true for individual form element, this can be extended to have in a separate service file for re-usability.

# **Tools and Libraries**

- Angular version: 10.0

- TypesScript: 3.9

- Rxjs: 6.5

- Jest – 26.0.13

- ngrx – 10.0

# **Unit Testing**

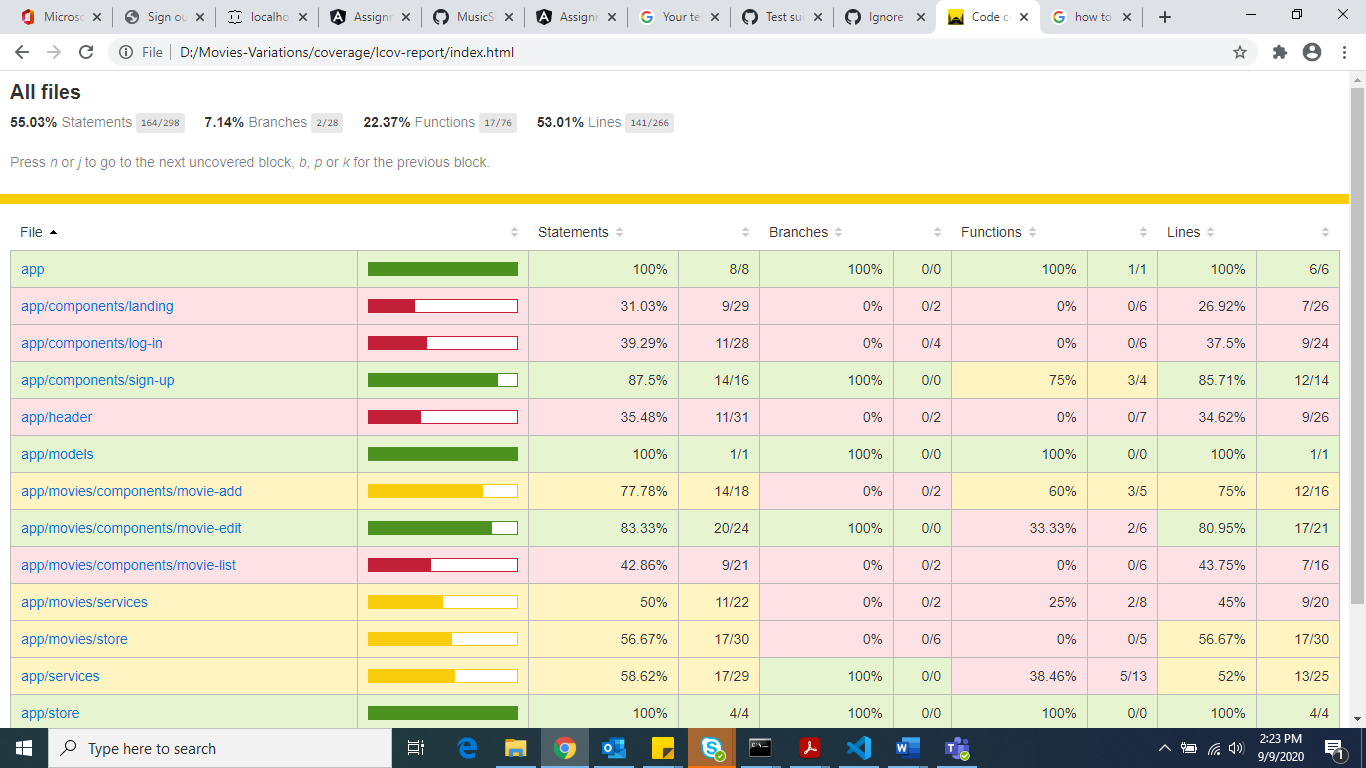
Added Unit Test cases for components.

It consists of one component-

* + header.component.spec.ts – It contain test cases for the all the logic for header component

**No of test cases - 6**

**Screenshot for unit test cases –**



# **Responsive**

* The application is responsive with all the device and desktop size.
* No Horizontal scrollbar in application.
* Used SCSS for designing.

# **Accessibility**

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

User can run application via keyboard and tab.

# **Theming**

I have done the theming for login page.

# **Setup**

1. Download and install node.js (version: 12.16)
2. Install npm and install @angular (version: 9.0)
3. Git clone <https://github.com/karthikakarthikeyan/TaviscaAssignment1> origin master
4. json-server --watch db.json
5. Navigate to cloned folder & run npm install
6. Run ng serve