**Images Sharing Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Image Sharing Application |
| Author | Monika Sawlani |
| Version | V1.0 |
| Creation Date | 07/09/2020 |
| Last Updated | 07/09/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc45989188)

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

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

[**3.** **Flow Chart** 5](#_Toc45989192)

[**4.** **Components** 6](#_Toc45989193)

[**5.** **Services** 6](#_Toc45989194)

**6. State Management**

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

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

[**9.** **Accessibility** 8](#_Toc45989199)

**10. Testing**

**11. Theme**

[**12.** **Setup** 9](#_Toc45989201)

# **Introduction**

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

# **Use-Case View**

The application use cases are -

1. **Image List** – It Display public listing of images and their associated data. for logged in users CRUD flow for Images and their info

* **Add Image to the List** – After adding list, image can be added inside a list. So, there will be a link to add a image with name and respective list.
* **Edit List –** Any image can be editable while clicking on edit name.
* **Delete Image** – Each Image will have a delete button, the image can be deleted by clicking on it.

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 Application |
| Storybook | Created for component |
| Theming | Created two themes for the app |
| Json-server | Used for Application |

# **Flow Chart**

New state

Header Page

State

Images

Render

Login

Handled By Dispatch

Delete Image

Edit Image

Add Image

# **4.Components**

* AppComponent – This component is used for Application Bootstrapping
* HeaderComponent- Header Page of application that show theming and all the image and their respective details.
* LoginComponent-This contain login for user.
* ImageComponent – Home Page of application that show all the image and their respective details with edit and delete feature and links for adding new image.
* AddEditComponent – This component contain form for add list and edit list.

# **Services**

There are two services –

1. AuthService – This service generally having logic about login and register
2. ImageService – This service generally having logic about image in the application and this contain code for getting image, edit card details, add new image and delete image logic.

# **State management:-** Ngrx for state management **.**

# **Validations**

1) Application covered the required validation for form fields.

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

# **Tools and Libraries**

- Angular version: 10

- TypesScript: 3.9.5

- Rxjs: 6.5

# **Accessibility**

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

User can run application via keyboard and tab.

1. **Created two themes for the app** – light and dark which can be changed from header

# **Jest Testing:-**

Added Unit Test cases for services.

It consists two services -

* + images.service.ts – Test cases are written in images.service.spec.ts for CRUD operation.
  + Auth.service.ts - Test cases are written in auth.service.spec.ts for login/signup
  + Reducer.ts - Test cases are written in images.reducer.spec.ts and auth.reducer.spec.ts

# **Open Item and Enhancement**

StoryBook

1. **SetUp**
2. Download and install node.js (version: 14.5.0)
3. Install npm and install @angular (version: 10)
4. Git clone <https://github.com/mgsawlani1/imagesSharingApp.git> origin master
5. Navigate to cloned folder & run npm install
6. Run ng serve
7. To run json-server -**json**-**server** --watch db. **json**

**Performance Scope**

