**Offers Management Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Offers Management Application |
| Author | Sneha Waje |
| Version | V1.0 |
| Creation Date | 07/09/2020 |
| Last Updated | 07/09/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc50421519)

[**1.** **Introduction** 3](#_Toc50421520)

[**2.** **Use-Case View** 3](#_Toc50421521)

[**3.** **Flow Chart** 4](#_Toc50421522)

[**4.** **Components** 5](#_Toc50421523)

[**5.** **Services** 5](#_Toc50421524)

[**6.** **Validations** 6](#_Toc50421525)

[**7.** **NgRX Store** 6](#_Toc50421526)

[**8.** **Tools and Libraries** 6](#_Toc50421527)

[**9.** **Unit Testing** 6](#_Toc50421528)

[**10.** **Responsive** 7](#_Toc50421529)

[**11.** **Accessibility** 8](#_Toc50421530)

[**12.** **Open Items** 8](#_Toc50421531)

[**13.** **Setup** 8](#_Toc50421532)

# **Introduction**

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

# **Use-Case View**

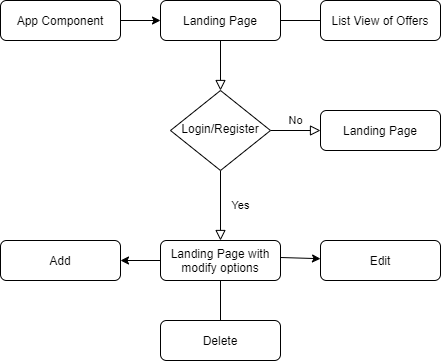
The application use cases are –

* **Login** – Allows user to login with credentials with provided form validations.
* **Register** – Allows user to register and login with credentials with provided form validations.
* **Show List of Offers** – Show list of all offers
* **Add Offer** – It provide functionality to add offer.
* **Edit Offer –** Any offer can be editable by clicking edit button provided in front of it by accessing ID of particular offer.
* **Delete Offer**– Any offer can be deleted by clicking delete button provided in front of it by accessing ID of particular offer.

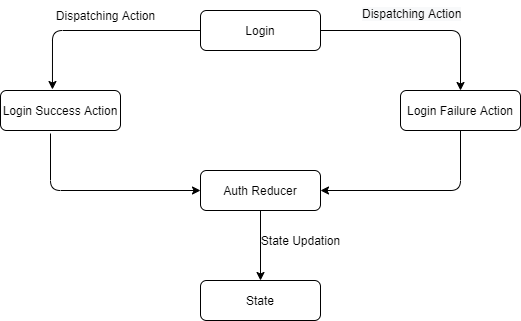
These use cases are initiated by the user.

# **Flow Chart**

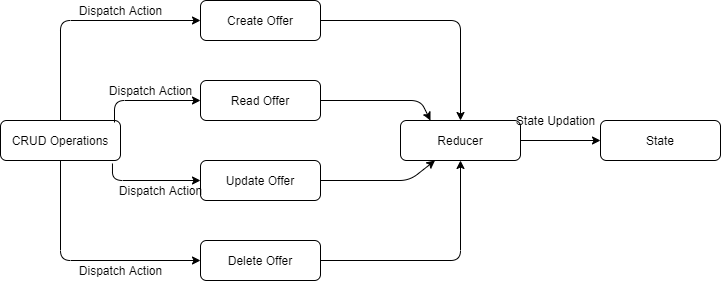
* **Basic Flow**



* **Login NgRX**



* **CRUD NgRX:**



# **Components**

* LandingComponent – This component is used for basic view of application with Login, Register, and list view option.
* LoginComponent – Login component allows a user to login with username and password and give access to modify list of offers.
* RegisterComponent – Register component allows a user to register with username and password and then give access to all functionalities accessed with login.
* AddOfferComponent – This component contain form to add new offer to the list and show it.
* AddOfferComponent – This component contain form to edit existing offer and show in the list.

# **Services**

There are two services –

1. AuthService – This service generally having logic about login and registration of a user.
2. OfferService – This service generally having logic about CRUD operations i.e. Create, Read, Update and Delete.

# **Validations**

1) Application covered the required validation for Username.

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

# **NgRX Store**

The application uses ngRx for state management for user and offers data.

# **Tools and Libraries**

- Angular version: 10.0

- TypesScript: 4.0.2

- Rxjs: 6.5.5

- Jest: 26.4.2 (This is a unit testing framework; it consists assertion and mocking support for writing test cases.)

- NgRX Store: 10.0.0

-tslint: 6.1.0

# **Unit Testing**

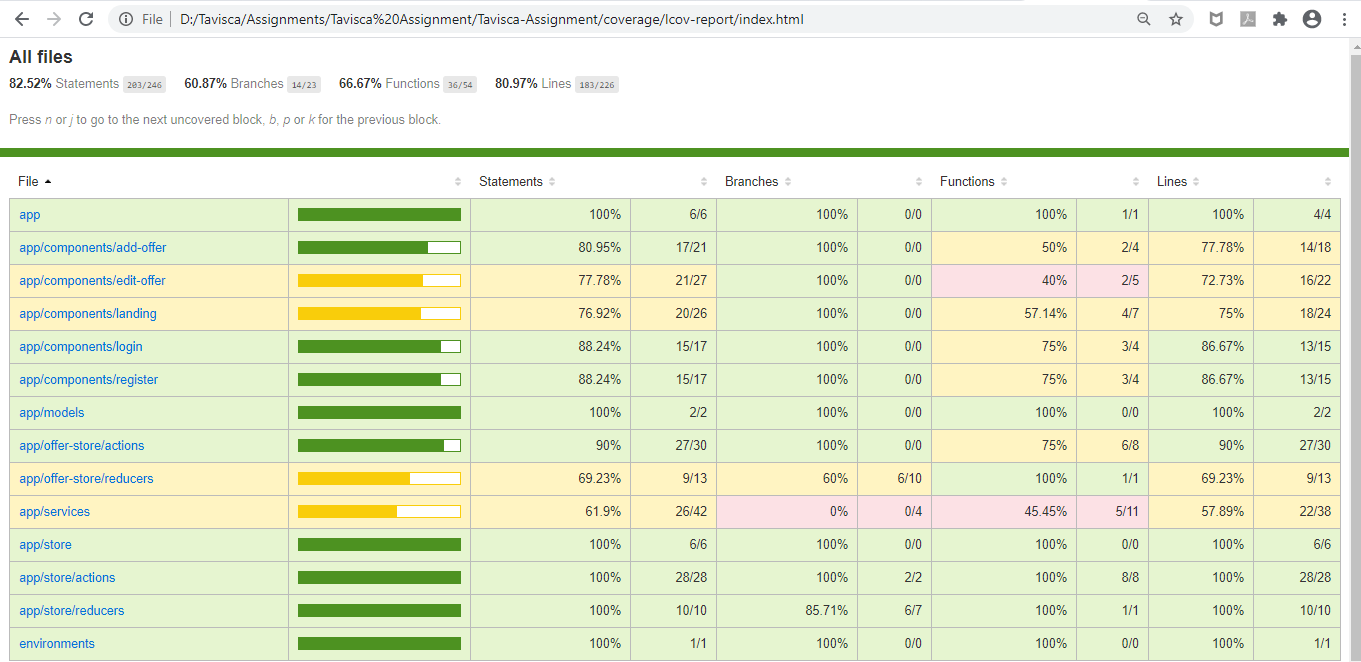
Added Unit Test cases for services.

It consists of two services –

* auth.service.spec.ts - It contain test cases for the all the logic like login and register user.
* offer.service.spec.ts - It contain test cases for the all the logic like add offer, update offer, delete offer and get list of offers.
* offer.action.spec.ts - It contain test cases for the all the logic like actions performed in offer components functionalities.
* offer.reducers.spec.ts - It contain test cases for the all the logic like actions performed in offer reducer functionalities.
* auth.action.spec.ts - It contain test cases for the all the logic like actions performed for login/register components functionalities.
* auth.reducers.spec.ts - It contain test cases for the all the logic like actions performed for login/register component functionalities.

**Total test cases: 37**

**Screenshot for unit test cases -**



# **Responsive**

* The application is responsive with all the device and desktop size.
* No Horizontal scrollbar in application.
* Used CSS grid for templating.

# **Accessibility**

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

User can run application via keyboard and tab.

# **Open Items**

* Web component.
* Lit-element
* I18
* Storybook

# **Setup**

1. Download and install node.js (version: 12.18.3)
2. Install npm(version:6.14.8) and install @angular (version: 10.0)
3. Git clone <https://github.com/snehawaje-1524/Tavisca-Assignment> origin master
4. Navigate to cloned folder & run npm install
5. To run API(JSON Server): npm run json:server
6. To run application: ng serve
7. For testing : npm run test-watch