**Quotes Sharing application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Quotes Sharing application |
| Author | Ruchira Rokade |
| Version | V1.0 |
| Creation Date | 09/09/2020 |
| Last Updated | 09/09/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc50408770)

[**1.** **Introduction** 3](#_Toc50408771)

[**2.** **Use-Case View** 3](#_Toc50408772)

[**3.** **Flow Chart** 4](#_Toc50408773)

[**4.** **Components** 5](#_Toc50408774)

[**5.** **Services** 6](#_Toc50408775)

[**6.** **Store** 6](#_Toc50408776)

[**7.** **Validations** 6](#_Toc50408777)

[**8.** **Tools and Libraries** 6](#_Toc50408778)

[**9.** **Unit Testing** 6](#_Toc50408779)

[**10.** **Responsive** 7](#_Toc50408780)

[**11.** **Accessibility** 7](#_Toc50408781)

[**12.** **Open Items** 7](#_Toc50408782)

[**13.** **Setup** 7](#_Toc50408783)

# **Introduction**

In this document, we can see the overview for the application which covers its basic flow and scenarios of the quotes sharing application.

It is mainly developed for managing quotes like performing CRUD operations.

Also there is a login and sign up functionality provided for the user.

# **Use-Case View**

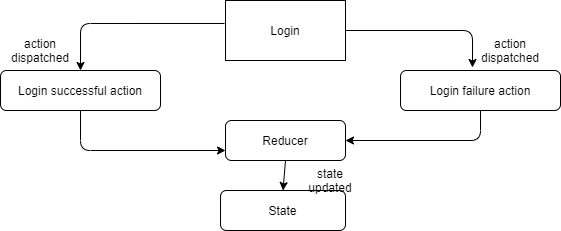
The application use cases for which user authentication is mandatory are -

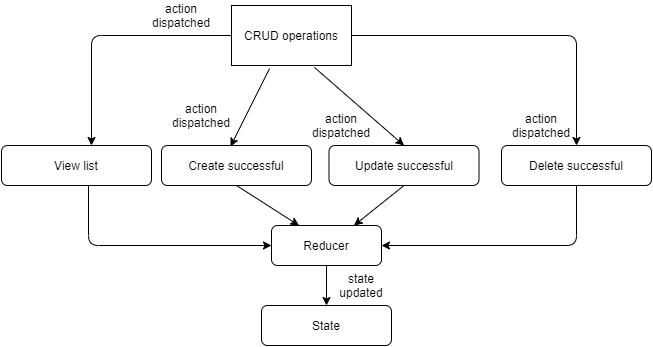
* **Add a quote** – The application provides the functionality for adding a new quote after user login.
* **Update a quote** – This functionality provides the facility to update the existing quote for the user.
* **Delete a quote –** The user has an option to delete the quote.

Use cases for which the user authentication is not mandatory.

* **List all quotes** -The user has the option to view all the quotes and their description.
* **Login**– The user can login into the application with his/her credentials.
* **Sign up** – If the user wants to modify any of the quote or create a new quote, he/she user can sign up, and then login to perform the CRUD operations.

# **Flow Chart**





# **Components**

* AppComponent – This component is used for Application Bootstrapping
* HomeComponent – This is the home page of the application
* QuoteListComponent – This is the component used to display the list of the quotes and from here the user can perform the actions related to the quotes.

From this component the user has an option to select the delete the quote.

* SignUpComponent – This component manages the signing up and login flow of the user
* CreateQuoteComponent – This component is used to manage the edit/create flow of the quote.

# **Services**

1. AuthService – This service takes care of the API requests for authentication of the user like login and sign up.
2. QuotesService – This service takes care of the API requests for CRUD operations of the quote.

# **Store**

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

# **Validations**

1)When an unregistered user tries to login, the application does not allow it to do so, first user needs to register/sign up then he can login.

2) Login and Sign up buttons are disabled until user enters all the fields.

3) Before deleting a quote the application asks for user confirmation.

**Validation strategy:**

- Used Reactive form for validation

- Used window.alerts and window.confirm for showing error messages from API and for confirmation resp.

# **Tools and Libraries**

- Angular version: 10.0

- TypesScript: 3.9.5

- Rxjs: 6.5.5

- Jest: 26.4.2 (This is a unit testing framework; it consists of tools/scripts to run the test cases.)

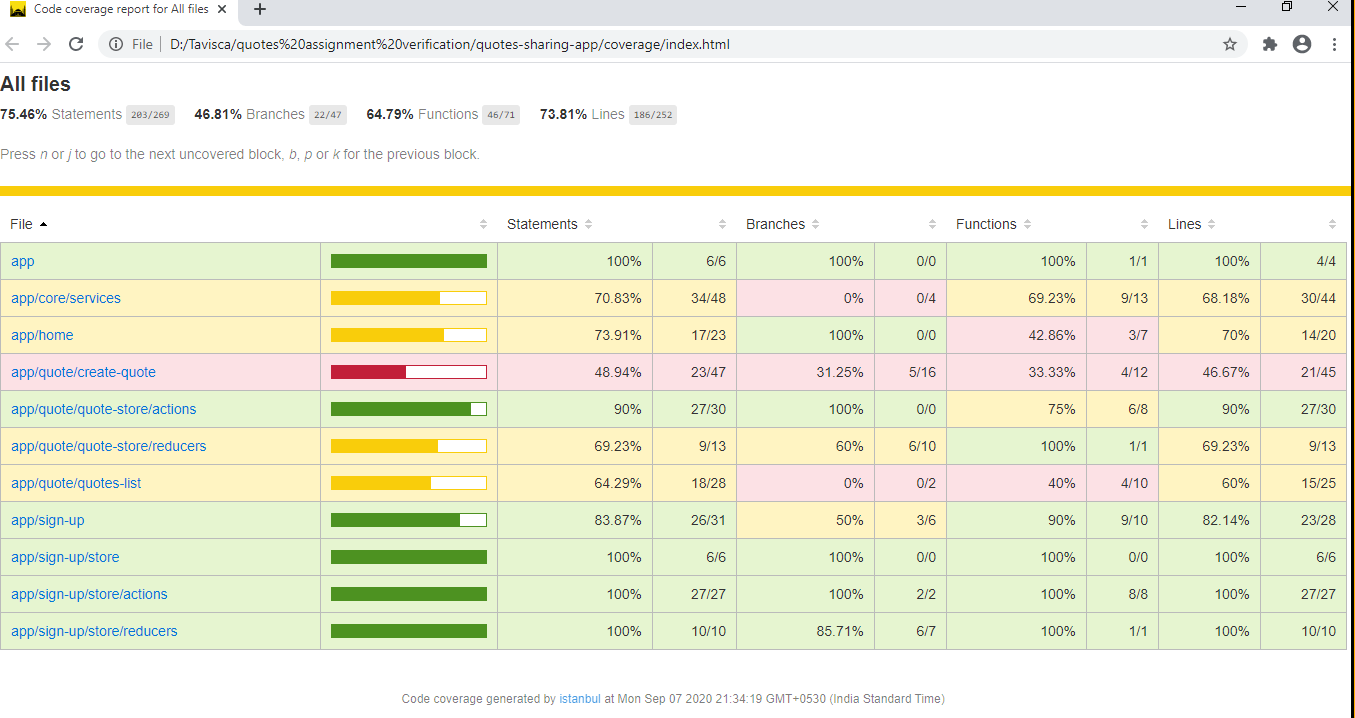
# **Unit Testing**

Added Unit Test cases for services.

It consists of following services -

* + Quotes.service.spec.ts – It contain test cases for the all the API calls for CRUD operations for quotes.
  + Auth.service.spec.ts - It contain test cases for API calls for the authentication.
  + User.action.spec.ts – It contains test cases for user actions
  + Quote.actions.spec.ts - It contains test cases for quote store
  + Quote.reducer.spec.ts – Contains the test cases for quote.reducer.ts
  + Auth.reducer.spec.ts – It contains test cases for auth.reducer.ts

**Screenshot for unit test cases -**



# **Responsive**

* Created responsive table which grows/shrinks in width according to the device.
* No Horizontal scrollbar in application.

# **Accessibility**

For Accessibility, the HTML markup contains aria-labels for buttons and tab indexes for input fields and table data.

# **Open Items**

1. Storybook

# **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/RuchiraRokade/Persistent-app origin master
4. Navigate to cloned folder & run npm install
5. Run ng serve
6. For testing use: npm run test
7. For running json server, goto server folder inside src\server\Json-server-Quotes
8. Perform npm install
9. Run command npm run json:server