**Technical documentation**

***Old MacDonald Application***

Date: Created by:

16.06.2019 Kiril Antonov Nikiforov

**Content:**

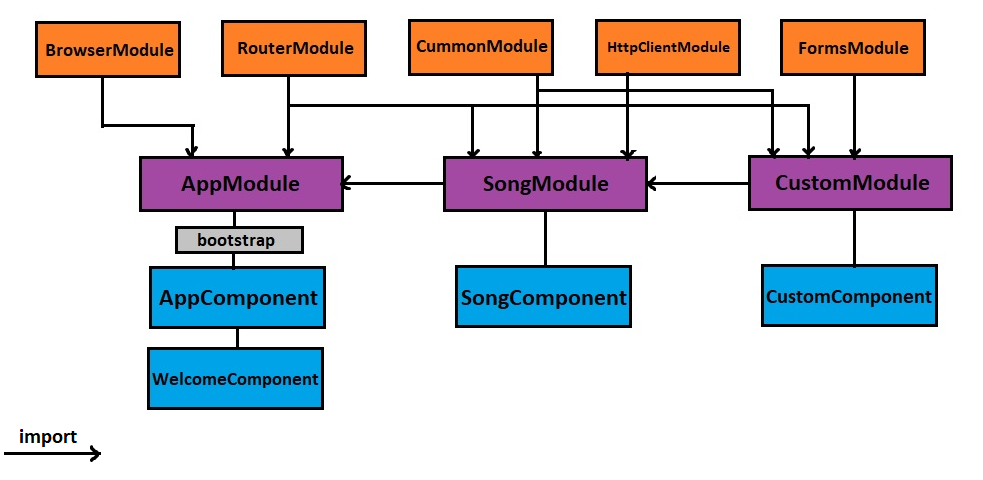
1. Used software
2. Application structure
3. Tasks specifics
4. Preparation and feedback

**Used software**

1. **Operation system:** Windows 10
2. **Source code editor:** Visual Studio Code v1.35.1
3. **Framework:** Angular CLI 8.0.3
4. **Script language:** TypeScript
5. **Additional:** Bootstrap 4.3.1 ,  
    Chrome 74.0.3729
6. **Other:** Some free to use pictures and icons

**Application structure**

1. **Diagram:**



1. **Explanation.**

We use three modules for our application, the main application module **AppModule** that imports **BrowserModule** and **RouterModule** needed for browser communication and navigation through our application ‘pages’ (It’s actually a single page application because we have only one index.html). We used **bootstrap** library to extend our views with standardized CSS. We have two components in this module: **AppComponent** that is out application main component that we use in every page and **WelcomeComponent** for our home page view and functionalities.

**SongModel** import **AppModule** (child -> parent relation) and also import **CummonModule** so it can use all of his parent functionalities and **HttpClientModule**, so we can use requests for XML data extraction. **CustomModule** is child of **SongModule** and import except the regular modules also **FormsModule**. We use this module for two-way data binding. Both modules have their expected main components: **SongComponent** and **CustomComponent**

**Tasks specifics**

1. **Main Tasks:**

First of all, I decide to use thee modules for my application. There isn’t much components to need such a separation but logically all modules can be expanded with additional modules and is easy to see them as separate pages. I tried to use standard name convection and structure for my application. Project folder and all components was created with CLI functionalities. I used TypeScript, because I’m more used to C# style of programming and strong typing.

I used json file (assets/animals.json) as my database that provides needed animals and their sound for the original song.

I use two-way data binding for solution of my second task.

I didn’t use much design for application, but add some pictures for flavour. All sources were free to use and added to assets folder.

1. **Unit tests:**

I don’t have much experience with unit tests so I didn’t create any custom ones, but used CLI ones and tried to meet their expectations. Afterword I fixed almost all of my problem except two karma failures that are related with HttpClientModule and FormsModule imports.

**Preparation and Feedback**

1. **Preparation:**

I finished beginner’s course for angular in [https://www.pluralsight.com](https://www.pluralsight.com/) by Deborah Kurata ,that help me establish my setup and gain needed information for my first angular project. I also used support information from forums and sites related with error problems that occur during development.

1. **Feedback:**

I really enjoy learning about and working with angular. It’s definitely modern and very powerful way of web development. It proofs its words that it is more than a framework. I will continue looking forward to learn and work with angular.