TATA CONSULTANCY SERVICES
Experience certainty.

Quiz Generator

An angular training

TATA

TATA CONSULTANCY SERVICES

Quiz Generator

Oct 2019
Version 1. 0

Authors:

Sergio, Salazar sergio.y.salazar@pwc.com

1. Document revision list

Version No / Date	Change Description	Page nos Reference (Current Version)	Page no Ref (Prev Version)
1.0 Oct 2019	*Initial Version*	All	NA

2. About this document

Purpose

This document serves as a guideline for an evaluation about Angular. The evaluation is centred on Angular App with architecture based on components and angular service consuming a Node micro service

Intended Audience

This document is primary intended for trainees focusing on learning Angular technology.

3. Contents

4. Introduction

Quiz generator is a frontend and backend application based on the Angular and Node.js framework, focused on providing its users with a tool to perform online evaluations about specific topics.

The application involve the development of the below items:

- One node.js micro service, which will use an input parameter to serve static json objects with questions, related to some specific technology. **This micro service needs to be developed along with this other front-end application.**

- One angular application that will consume the micro service to get the proper json object to do the quiz, and will serve html content in order to complete the application workflow. To achieve that, the angular app (quiz manager UI) should start with landing page that allow user to enter their login and password, then processing that information to serve the corresponding quiz options, and showing at the end an information message.

5. Problem Definition and Requirements

5.1 Application Architecture

The architecture of the application should be designed using as much as possible/needed, a folder with all the components, and other folder with a service that will consume the node.js micro service and other with all the pages of the app.

5.1.1 Client side tools

In order to test the application, a web browser must be used.

5.2 Application Requirements

5.2.1 General

Create a quiz generator trough a micro service (provided, do not needs to be created) that will serve static json objects associated to some specific area, and trough another angular application (quiz manager UI) that will invoke the micro service. The Angular piece of quiz manager UI must render html content in order to provide a user interface to follow the application workflow.

5.2.2 Specifics

- The incoming json parameter will indicate to the micro service which specific json object this will serve, based on the required area. Please refer to the micro service code to see the expected payload.

- The angular app must have a Login page with:
  - ➢ Header with title and image on the corner of the app.
  - ➢ Textbox to write the user name.
  - ➢ Textbox to write the user password.
  - ➢ Error message if the credentials are not corrects.
  - ➢ Bottom to enter the app.
  - ➢ The credentials for that should be stored in a json file and the validation must be performance against this file.

- The angular app must have a Home page with:
  - ➢ Header with title and image on the corner of the app, also with the name of the user that is login.
  - ➢ Select with all the names of each area.
  - ➢ Bottom to send the info (name of user, name of area)

- Once the user click on submit, the angular app must show a Quiz page with:
  - Header with title and image on the corner of the app, also with the name of the user that is login.
  - For each question a label and radio bottoms to select an option. (Minimum 5 questions)
  - Bottom to send the info (number of question, answer and user).

- Once the user sends the info, the angular app must show in a new page or in the same one, how many answers are correct and incorrect, and offer to the users a mechanism that allow them to exit of the app or start a new quiz.

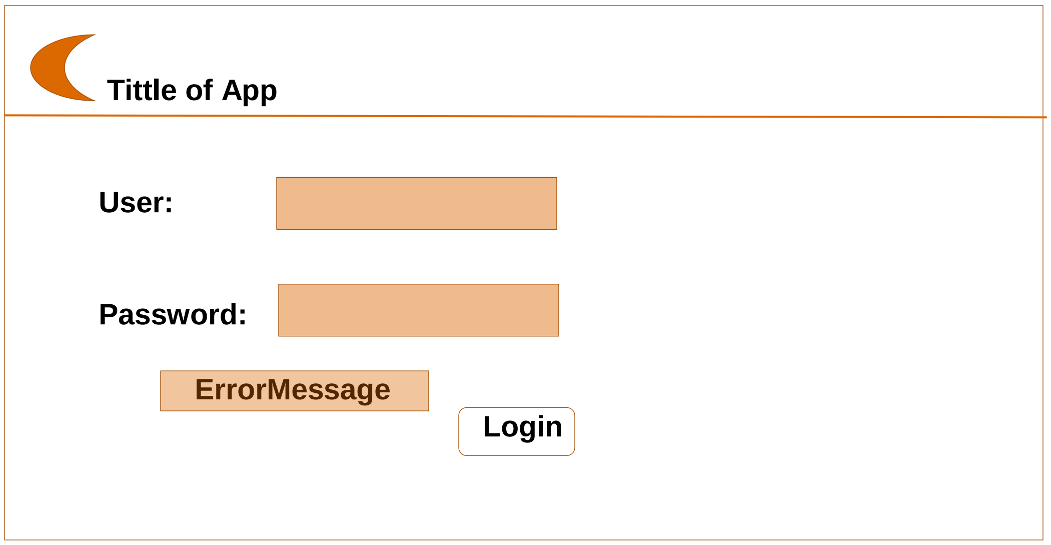
6. Design

6.1 Interface

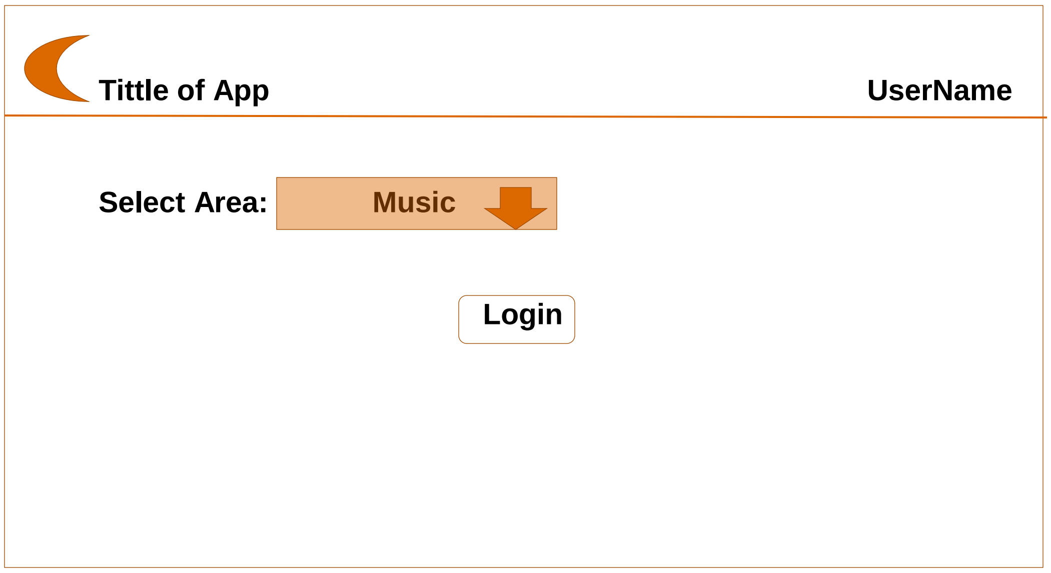
The user interface must be flexible and user friendly.
You can use bootstrap or angular material to represent all the styles.

Following are an examples of each template:

- **Login:**



- **Home Page:**

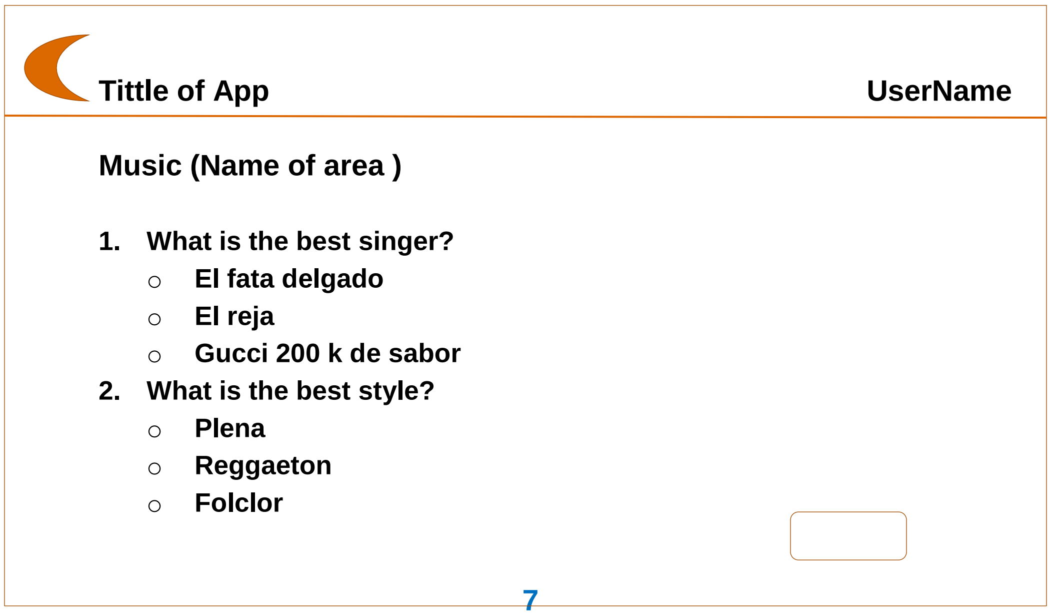


Tittle of App **UserName**

Select Area: **Music**

Login

- **Quiz Page:**



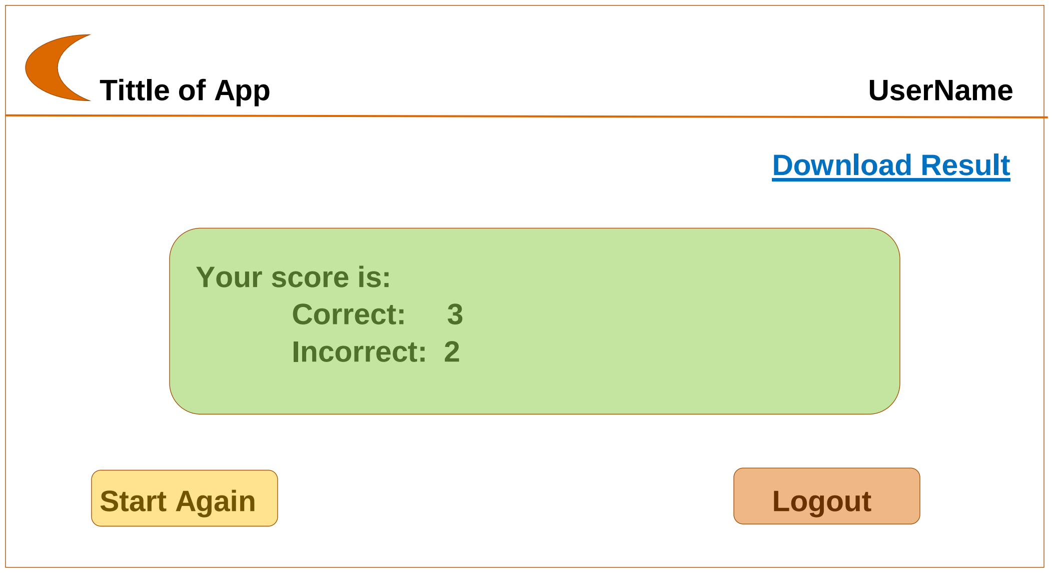
Tittle of App **UserName**

Music (Name of area)

1. **What is the best singer?**
   - **El fata delgado**
   - **El reja**
   - **Gucci 200 k de sabor**
2. **What is the best style?**
   - **Plena**
   - **Reggaeton**
   - **Folclor**

Submit

- **Final Page:**



Tittle of App

UserName

Download Result

Your score is:
Correct: 3
Incorrect: 2

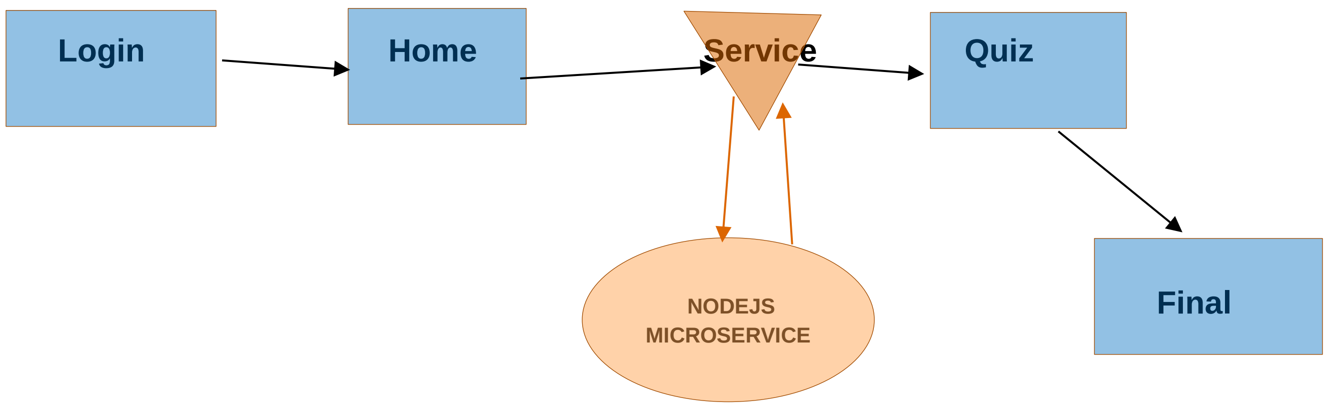
7. Required capabilities.

The app must have all the followings capabilities of the Angular Framework:

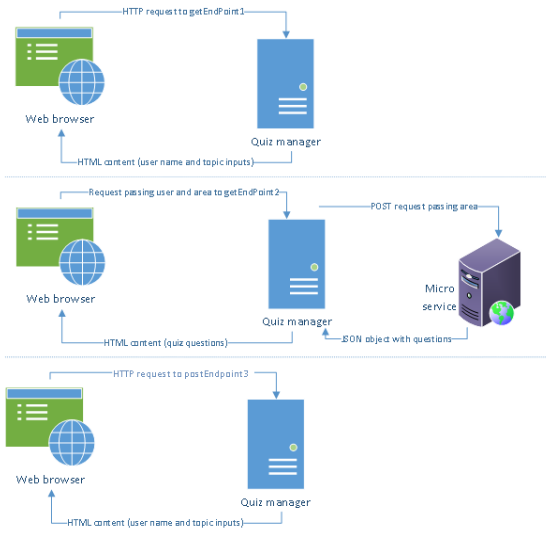
- Model driven forms.
- Template driven forms.
- Service with injection.
- Pipes.
- Directives.
- Data binging,
- Handling of routing and navigation.
- Observables.
- Minimum two modules.

8. High-level diagram.

Basically the architecture and structure of the app is the follow:



Micoreservice:


HTTP request to getEndPoint1
Web browser
Quiz manager
HTML content (user name and topic inputs)
Request passing user and area to getEndPoint2
POST request passing area
Web browser
Quiz manager
Micro service
HTML content (quiz questions)
JSON object with questions
HTTP request to postEndpoint3
Web browser
Quiz manager
HTML content (user name and topic inputs)

9. Extras

Extras are a required part of the application. Before starting, the trainee should select at least two of the available improvements below, this must to be implemented after finishing the core application.

9.1 Logs reporting

The quiz generator should provide, in somehow, an easy way to report in excel or csv format the final report with all the question correct and incorrect and the user can download it.

9.2 Timer.

The quiz manager UI must have a timer (component) when the user start the quiz, if the user has not yet made the request, the bottom will unavailable and the app should show an pop up message.

9.3 Animation package

The quiz manager UI should implement some features of the Angular animations package. Animations can improve your app and user experience.

9.4 Application log-out

In the header of the application, add a “sign-off” link that allow to the users to be logged out of the application, and that redirect to a signed off page.

9.5 Show the logged user

Since the users must enter their user and password to go inside the quiz manager UI, could be great if we can see the user that is logged in at the top of every page.