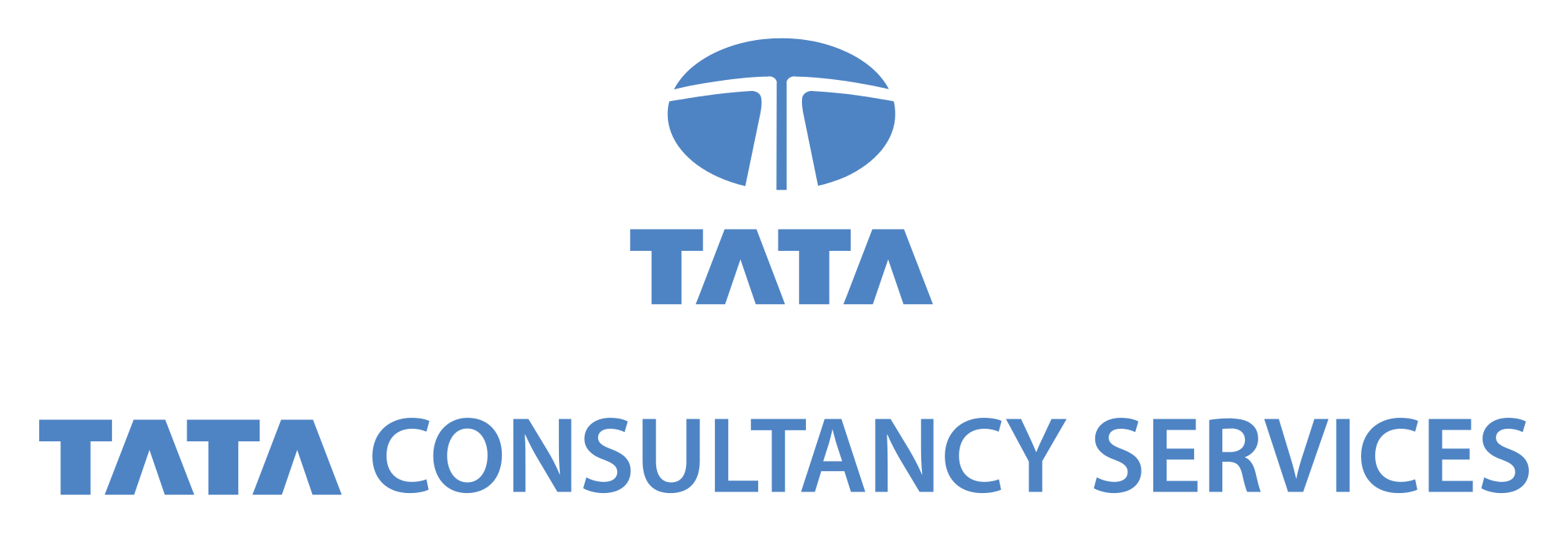
Car Rental

An angular training



|  |
| --- |
|  |
| Car Rental  *Nov 2019*  *Version 1. 1* |

Authors:

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

Alberto, Machuca alberto.x.machuca@pwc.com

Humberto, Rodrigues humberto.pereira@pwc.com

# Document revision list

|  |  |  |  |
| --- | --- | --- | --- |
| Version No /  Date | Change  Description | Page nos Reference (Current Version) | Page no Ref (Prev Version) |
| 1.0 Feb 2019 | Initial Version | All | NA |
| 1.1 Nov 2019 | Changes |  |  |

# 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 Json file through a service.

## Intended Audience

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

# Contents

[1. Document revision list 2](#_Toc23460011)

[2. About this document 2](#_Toc23460012)

[Purpose 2](#_Toc23460013)

[Intended Audience 2](#_Toc23460014)

[3. Contents 3](#_Toc23460015)

[4. Introduction 4](#_Toc23460016)

[5. Problem Definition and Requirements 5](#_Toc23460017)

[5.1 Application Architecture 5](#_Toc23460018)

[5.1.1 Client side tools 5](#_Toc23460019)

[5.2 Application Requirements 5](#_Toc23460020)

[5.2.1 General 5](#_Toc23460021)

[5.2.2 Specifics 5](#_Toc23460022)

[6. Design 6](#_Toc23460023)

[6.1 Interface 6](#_Toc23460024)

[7. Required capabilities. 8](#_Toc23460025)

[8. High-level diagram. 9](#_Toc23460026)

[9. Extras 9](#_Toc23460027)

# Introduction

Car rental is a front-end application based on the Angular framework using Typescript, focused on providing its users with an app to see the cars to reserve.

The application involve the development of the below item:

* One angular application that will consume the json file with information of cars and customers to be rented, and will serve html content in order to complete the application workflow. To achieve that, the angular app (car rental) should start with landing page that allow user to enter their login and password, then processing that information to serve the corresponding car module (see information of all cars and its details) and customers module (a form to register a new customer).

# 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 json file, other with all the pages of the app ant other with the models

### 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 an angular app that will consume static json objects associated to information of cars rented and information of customers. The Angular piece of car rental must render html content in order to provide a user interface to follow the application workflow.

5.2.2 Specifics

* The angular app must have a Login page with:
* Header with title and image on the corner of the app and the current date.
* 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.
* Footer with information of the app.
* The credentials for that should be stored in the local storage
* The angular app must have a Home page with:
* Header with title and image on the corner of the app and the current date.
* Two options to enter a cars module and hhcustomer’s module.
* Footer with information of the app.
* Once the user click on cars module, the angular app must show a cars page with:
* Header with title and image on the corner of the app and the current date.
* A list with all the cars with information about each one: mark, model, status, image and a button “History Reservation” to see all the reservations of that car.
* Once the user make click on “History Reservation” button, the app show all the history reservation (date start, date end and customer’s name) of it in other list.
* Once the user click on customer module, the angular app must show a customer page with:
* A form with the following fields to register a new customer:
* First and last name (minimum 4 characters and required)
* Email (validation of @ and required)
* Telephone (number field with 8 characters and required)
  + Button “register” (without logical implementation)

# 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:**

**Tittle of App Date  
**

**User:**

**Password:**

**ErrorMessage**

**Login**

**Footer**

* **Home Page:**

****

**Tittle of App Date**

**Cars Customers**

**Footer**

* **Cars Page:**

****

**Tittle of App Date**

****Mark **History Reservation:** Car idModel Status **Customer Name  
 History Date start Date end**

****Mark **Customer Name**Model **Date start Date end**Status **History**

**Footer**

* **Customer Page:**

****

**Tittle of App Date**

**Register customer**

First name:

Last name:

Email:

Phone:   
  
 Error message Register

Footer

# 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.

# High-level diagram.

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

**Login Home Car Page** Service

**Customer Page** **JSON**

# Extras

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

#### 9.1 Show the name of the user in the header component.

When the user enter their credentials, his name should be show in the header component.

#### 9.2 Show customer details.

Create a json file with customer’s information and in the customer page or in a new page show the same in a list.

#### 9.3 Handling routes and log-out

Handle the url routes so if the user is not logged, he can’t put directly the url of any route of the app on the browser, and then the app will show a message error. And 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 login page.

#### 9.4 Implement reservation

In the car page for each car that is not reserved implement a button to reserve it (without details of the customer and date of reserve). The status of that car will change and reflect it in the moment.

#### 9.5 Implement reservation details

If you selected the previous extra and want to improve it, when the user clicks on the reservation button, open a new component to enter the name of customer and date start/end of that reservation.