Customer Management Portal

# Objective:

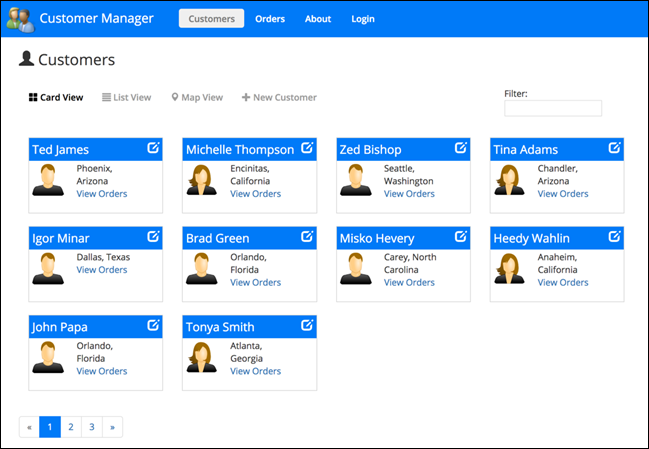
To build the UI of a customer management portal which can list the customers for a specific region and their order details. Application to include login screen to authenticate the users and to authorize the roles – Customer Manager, Agent and user admin.

# Functional Requirements:

Application should include the below functional flows:

## View Customers

* Upon login, customer manager should be able to view the existing customers
* View of the customers can be displayed in Card View or in List View as shown in below screenshot.



* Include search criteria for listing the customers and include pagination as well.
* Both customer name and view orders should be displayed as link to drill down the detailed information.

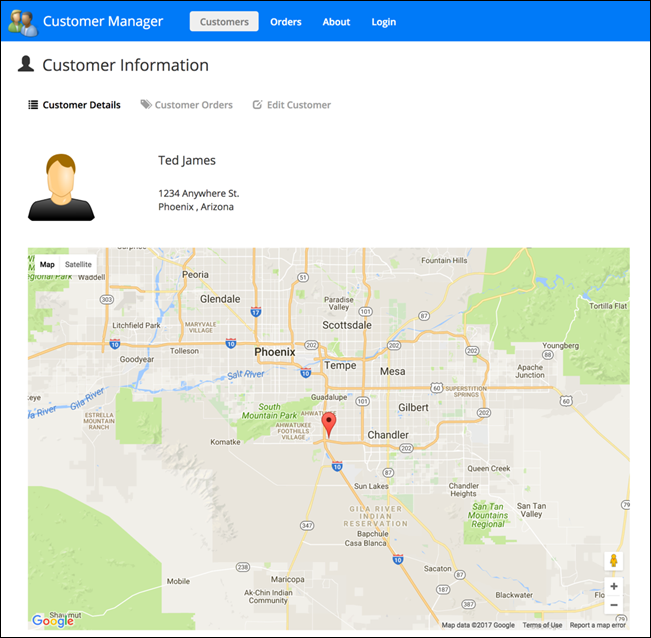
## View Customer detail (single customer info) and customer orders

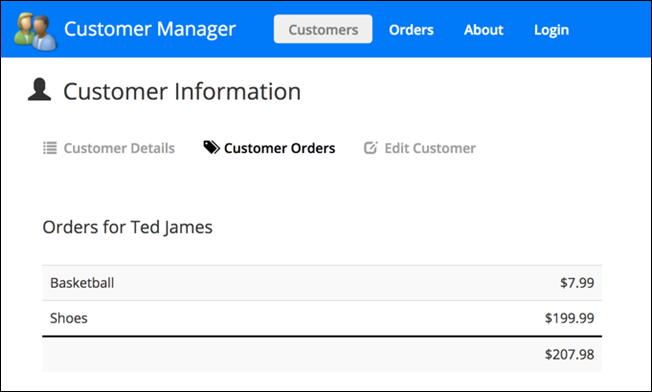
* **Display the details of the customer that may include**

1. Full name
2. Address
3. Contact number
4. Account details
5. Image (if available)
6. Other info

* **Orders by the customer**

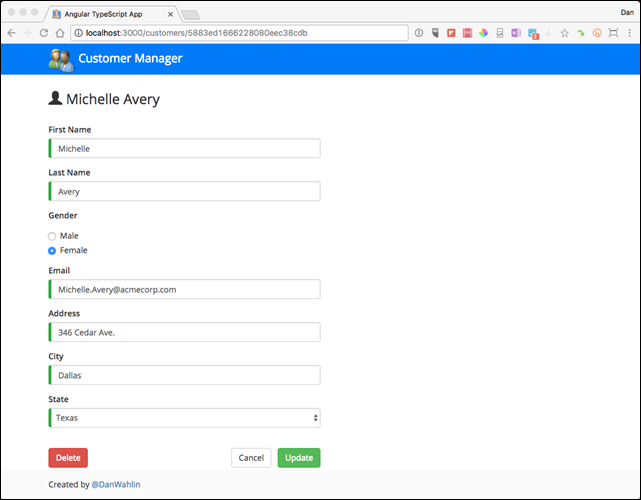
1. Display all completed, pending orders with amount and date info
2. For pending order, display the shipment details and the target delivery date





1. Create New Customer & Edit Customer

* Both Create and Edit customer details may look similar, but edit screen should display the values fetched from the backend.
* Only Customer manager should be able to create and edit customer details.
* Agent should be able to create customer details but **not edit** the details.
* Display the name and role of the user on top of the page



1. Include Login & admin screen as well

# Technical Requirements:

* Above attached images are just for reference only. Themes and styles can be applied in tune with latest trends.
* Include UI components such as navigation bars, tabs, header, footer and side bar as per your preference.
* Include Authentication Guard, to authorize based on the roles. For example, Agent should not be able to access the Edit screen.
* For Create & editing customers, include template driven or reactive forms.
* Application to include the below technical features:

1. Single Page Application features
2. Using NgModule to organize applications and support lazy loading
3. Data binding syntax
4. The Role of components
5. Input and Output properties of Component Interaction
6. component life-cycle
7. directives in templates
8. Creating injectable services
9. Observables and Promises
10. Http and Observables to interact with RESTful services
11. Working with pipes
12. Build and use custom pipes
13. Routes and Navigations
14. Route Guards
15. template-Driven and reactive forms

* Use a mock service for storing and displaying the data. Need not be with actual DB or real backend service.