|  |  |
| --- | --- |
| Use Case Name | Employee Management |
| Domain | - |
| Complexity | Medium |

Table of Contents

[1. Business Objective 3](#_Toc15030972)

[2. Functionality 3](#_Toc15030973)

[3. Scope and Limitations 3](#_Toc15030974)

[4. Design Considerations / Assumptions: 3](#_Toc15030975)

[5. UI Screens / Pages 5](#_Toc15030976)

[a. Main Page 6](#_Toc15030977)

[b. Employees List Page 7](#_Toc15030978)

[c. Add Employee 9](#_Toc15030979)

[d. Edit Employee 11](#_Toc15030980)

[e. Employee Details 12](#_Toc15030981)

[6. Application Flow 14](#_Toc15030982)

# Business Objective

Employee Management application helps to manage the employees of an organization.

Below are the Functionalities that customer is enabled with.

* Main Page
* Employees List page
* Add Employee
* Edit Employee
* Delete Employee
* Employee details

# Functionality

This application will provide a facility to manage the employees. Managing includes create, read, update and delete (CRUD). Existing employees list can be viewed. Employees can be filtered based on the starting letter of the name. Adding new employees is also possible. If the details about an employee changes, say employee has been moved to a different location, then it can be modified. If the employee leaves the organization, employee can be deleted from the list of employees. Also, if details about a specific employee is needed, that also can be shown.

# Scope and Limitations

Application right now is doing all the CRUD operations.

# Design Considerations / Assumptions:

1. You can store the data in angular-in-memory database
2. [http://host:port/contextroot/{path}](http://host:port/contextroot/%7bpath%7d)

In the above URL**,** *contextroot* isthe Context root of the application. You can use any value for contextroot and port in your local environment during development but do not **hardcode** them in the **application** source code, as the system will replace it with the dynamically generated name as context root and deploy in the application server to validate the functional test cases using selenium web driver **API**. In case if you are not using the context root **(say for example, Dot Net Core and NodeJS applications may not really have the context root**), it will be replaced with **blank**.

For example:

Java: <http://localhost:8080/empmanagement/login>

NodeJS: <http://localhost:3000/login>

Dot Net: <http://localhost:5001/login>

1. Do not refer any third-party libraries (CSS/JS libraries) directly from internet in the code, instead download them and use as local reference.
2. Variable **Identifier** of HTML should be id/name for input types, id for <div> and anchor text for <a> tags. It should be followed strictly.
3. **Do not hard code port number and context root anywhere in the code**
4. For location dropdown values, please use below data.
   1. Bangalore
   2. Chennai
   3. Pune
   4. Hyderabad
5. Below list of employees must be stored as initial employees list in angular-in-memory database. If any new employee is added by using “Add Employee” screen, the “id” must be incremented by 1.

let employees = [

{

id: 1,

name: "Ram",

location: "Bangalore",

email: "ram@mail.com",

mobile: "9867512345"

},

{

id: 2,

name: "Raj",

location: "Chennai",

email: "raj@mail.com",

mobile: "7867534521"

},

{

id: 3,

name: "Vinay",

location: "Pune",

email: "vinay@mail.com",

mobile: "9975287450"

}

];

# UI Screens / Pages

**Please read the below points carefully before you start the development of the use case.**

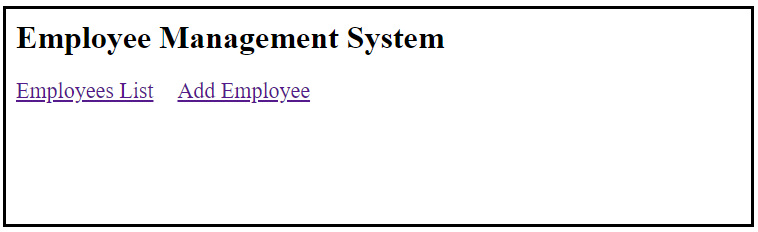
Each page should have the following:

* Title of the page
* All the input elements to capture the user input
* Buttons/Links for any action like submit, cancel etc.,
* **All the HTML elements should have id or name attribute**
* **Element id or name should be unique and is case sensitive. If you use different names, the automated functional test cases will fail.**
* Success or Error message to be displayed should be wrapped in a <div> tag with the documented id/name. Please note the div id name is case sensitive.
* **Please use the success / error messages documented as is, otherwise the automated functional test cases will fail.**
* **Please use the** [**design considerations / assumptions section**](#_Design_Considerations_/) **to configure the master data, without which the functional tests will fail.**
* The colours that appear in the sample screen-shots of the views below are only indicative. You can choose to ignore using background colours for pages as it is **Optional** and in no way effects the functionality or evaluation of the application.

## Main Page

This page is the main page shown when the application is launched.

Sample Home page UI:

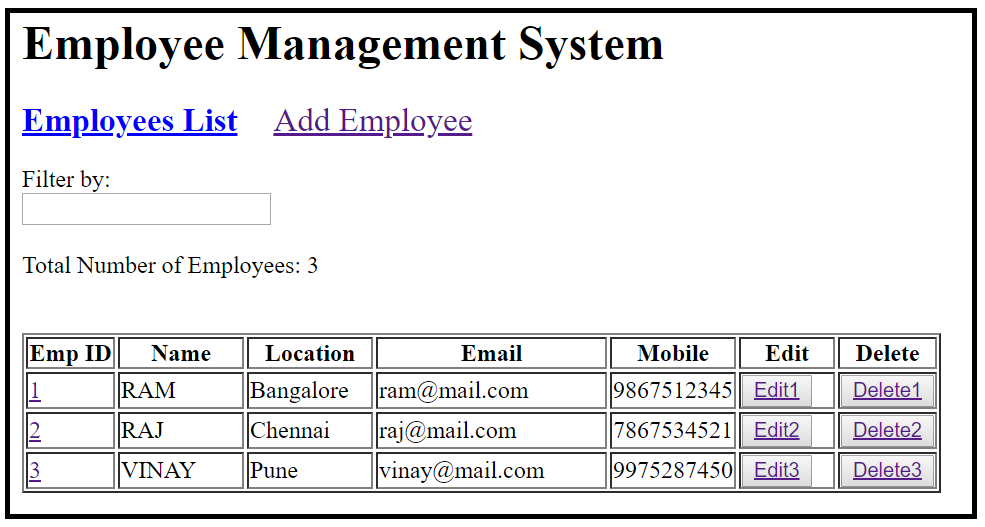


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field / Page attribute** | **HTML Element Type** | **Required** | **Identifier** | **Validation Message** | **Remarks** |
| URL | *http://host:port/contextroot/* | | | | |
| Title | Employee Management System | | | | |
| Heading | h1 | Yes | Heading of the application – Employee Management System | | |
| Employees List | Link | Yes | Link text = ”Employees List”  Ex: <a href=””>Employees List</a> | | On click, display the employees list (Employees list page) |
| Add Employee | Link | Yes | Link text = ”Add Employee” | | On click, take the user to add employee page |

## Employees List Page

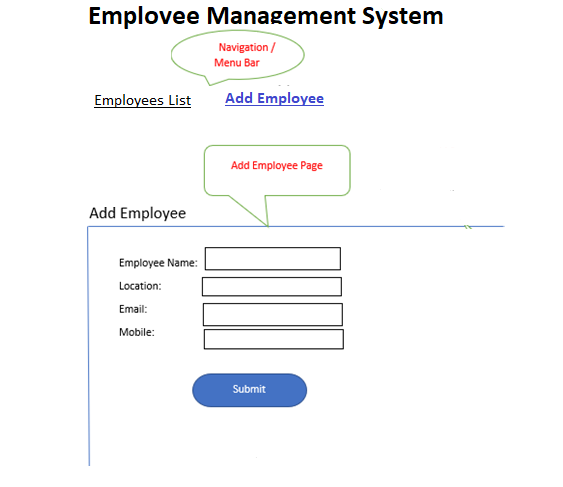
When user clicks on ”Employees List” link, below page is shown. On page load, the initial set of employees should be displayed. Employees list appears as in the initial list, newly added employee will come at the end of the list. Id for the new employee will be incremented by one. Display the results sorted by Emp Id, ascending order.

Sample page looks like below.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field / Page attribute** | **HTML Element Type** | **Required** | **Identifier** | | **Validation Message** | **Remarks** |
| URL | [http://host:port/contextroot/employees](http://localhost:4200/test/employees) | | | | | |
| Title of the page | Employee List | | | | | |
| Employee List | Label / Text | Header Text | | | | |
| Filter By | Label |  |  |  | |  |
| Filter By | Text Box |  | id=”filter” |  | | When user types in a character. Use that character to filter the employees based on the first letter of the name. Example: If filter text is v OR V, Employee list only should contain Vinay (Not Ram and Raj). If empty, display all the employee. If a character is entered and there is no matching employee, then the list will be empty. |
| Total Number of Employees | Label |  |  |  | |  |
| Number of Employees | Label value | Yes | Id=”count” |  | | Count should display the number of employees displayed on page |
| Emp ID | Column Header Text. | | | | | |
| Name | Column Header Text | | | | | |
| Location | Column Header Text | | | | | |
| Email | Column Header Text | | | | | |
| Mobile | Column Header Text | | | | | |
| Edit | Column Header Text | | | | | |
| Delete | Column Header Text | | | | | |
| Emp ID | Link | Yes | Link text should be **employee id**, like  Link text = ”1” for row 1  Link text = ”2” for row 2 etc. | | | Table data to be displayed in ascending order of emp id |
| Name |  | Yes | id=”name1” for row 1  id=”name2” for row 2 etc. | | |  |
| Location |  | Yes | id=”loc1” for row 1  id=”loc2” for row 2 etc. | | |  |
| Email |  | Yes | id=”email1” for row 1  id=”email2” for row 2 etc. | | |  |
| Mobile |  | Yes | Id=”mob1” for row 1  Id=”mob2” for row 2 etc. | | |  |
| Edit | Link | Yes | Link Text = ”Edit1” for Employee1  Link Text = ”Edit2” for Employee2 etc. | | | On click, take the user to Edit Employee screen |
| Delete | Link | Yes | Link Text = ”Delete1” for Employee1  Link Text = ”Delete2” for Employee2 etc. | | | Remove the employee from the list and refresh the display results |

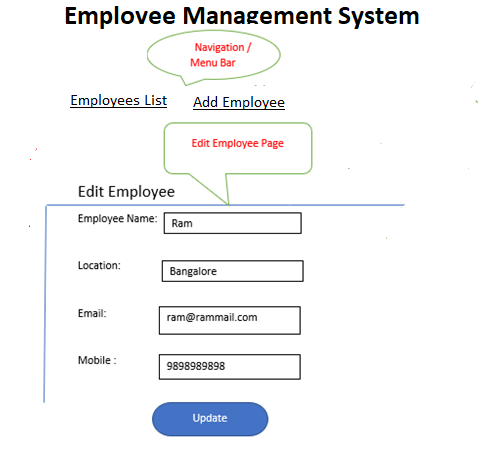
## Add Employee



All text fields should not allow more than 30 characters.

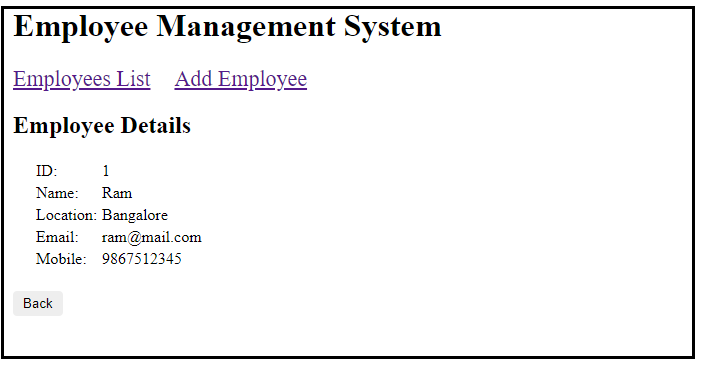
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field / Page attribute** | **HTML Element Type** | **Required** | **Identifier** | **Validation Message** | **Remarks** |
| URL | [*http://host:port/contextroot/*addEmployee](http://localhost:4200/test/addEmployee) | | | | |
| Title | Add Employee | | | | |
| Employee Name | Label |  |  |  |  |
| Employee Name | Text Box | Yes | id=”name” |  | Minimum length is 3.  Allows only alphabets. |
| Location | Label |  |  |  |  |
| Location | Select / dropdown | Yes | id=”location” |  | Minimum length is 3.  Allows only alphabets. For dropdown values, please refer design considerations section. |
| Email | Label |  |  |  |  |
| Email | TextBox / email | yes | id=”email” |  | Should be in email format. |
| Mobile | Label |  |  |  |  |
| Mobile | Text | Yes | id=”mobile” |  | Number of characters should be 10. Allows only numbers. |
| Submit | Button |  | id=”add” |  | Submit/add button should be disabled, if the input data is not valid as per the requirements described above.  For valid data, clicking on the button should add the employee with the provided details and should navigate to Employees List Page. New employee must be seen now in the Employees list. Each time a new employee is added, id is incremented by one. |

## Edit Employee



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field / Page attribute** | **HTML Element Type** | **Required** | **Identifier** | **Validation Message** | **Remarks** |
| URL | [*http://host:port/contextroot/*editEmployee](http://localhost:4200/test/editEmployee) | | | | |
| Title | Edit Employee | | | | |
| Employee Name | Label |  |  |  |  |
| Employee Name | Text Box | Yes | id=”name” |  | Minimum length is 3.  Allows only alphabets. |
| Location | Label |  |  |  |  |
| Location | select | Yes | id=”location” |  | Minimum length is 3.  Allows only alphabets. For dropdown values, please refer design considerations section. |
| Email | Label |  |  |  |  |
| Email | Text | Yes | id=”email” |  | Should be in email format. |
| Mobile | Label |  |  |  |  |
| Mobile | Text | Yes | id=”mobile” |  | Number of characters should be 10. Allows only numbers. |
| Update | Button |  | id=”update” |  | If the input data is not valid as per the requirements described above, disable the update button.  For valid data, clicking on the button should update the employee data with the provided details and should navigate to Employees List Page. The updated data should be displayed in the employee list page. |

## Employee Details



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field / Page attribute** | **HTML Element Type** | **Required** | **Identifier** | **Validation Message** | **Remarks** |
| URL | [http://host:port/contextroot/details](http://host:port/test/details) | | | | |
| Title | Employee Details | | | | |
| ID | Label |  |  |  |  |
| ID | Label value | Yes | id=”id” |  |  |
| Name | Label |  |  |  |  |
| Name | Label value | Yes | id=”name” |  |  |
| Location | Label |  |  |  |  |
| Location | Label value | Yes | id=”location” |  |  |
| Email | Label |  |  |  |  |
| Email | Label value | Yes | id=”email” |  |  |
| Mobile | Label |  |  |  |  |
| Mobile | Label value | Yes | id=”mobile” |  |  |
| Back | Button |  | id=”back” |  | Clicking on button, display “Employees List” page |

# Application Flow

|  |  |  |
| --- | --- | --- |
| **Current Page** | **Action** | **Next / Landing Page** |
| Main Page | Click on “Employees List” link | Employees List Page |
| Main Page | Click on “Add Employee” Link | Add Employee Page |
| Employees List Page | Click on “Edit” link | Display Edit Employee Page for selected employee |
| Edit Employee Page | Click on “Update” button | Employees List Page with updated details |
| Add Employee Page | Click on “Submit” button | Employees List Page with newly added employee |
| Employees List Page | Click on respective Emp ID link, eg “1” to see details of that employee | Display employee details page |
| Employee details Page | Click on “Back” button | Employee List Page |
| Employees List Page | Click on “Delete” link | Delete the employee and display employees list page with updated list of employees. |
| Employees List Page | Enter the filter criteria | Refresh the employee list with filtered data |