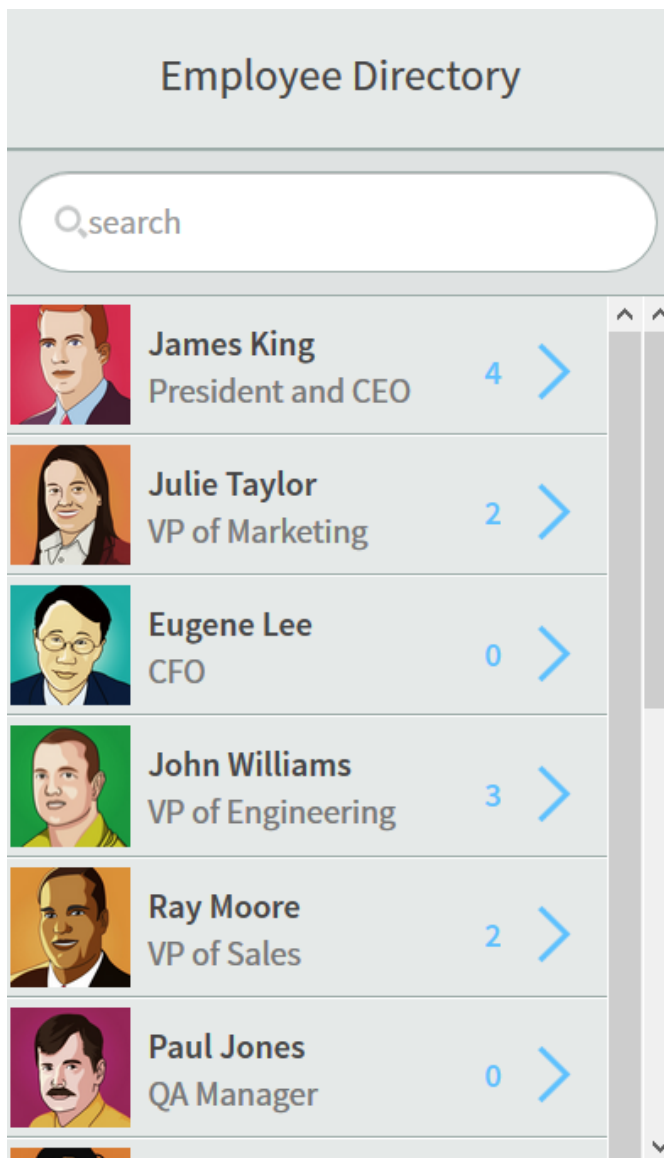


Project #2 Employee Management with MEAN Stack

This project is to develop a MEAN application (with nodeJS and either mySQL or MongoDB as backend) for employee directory. It should have following pages:

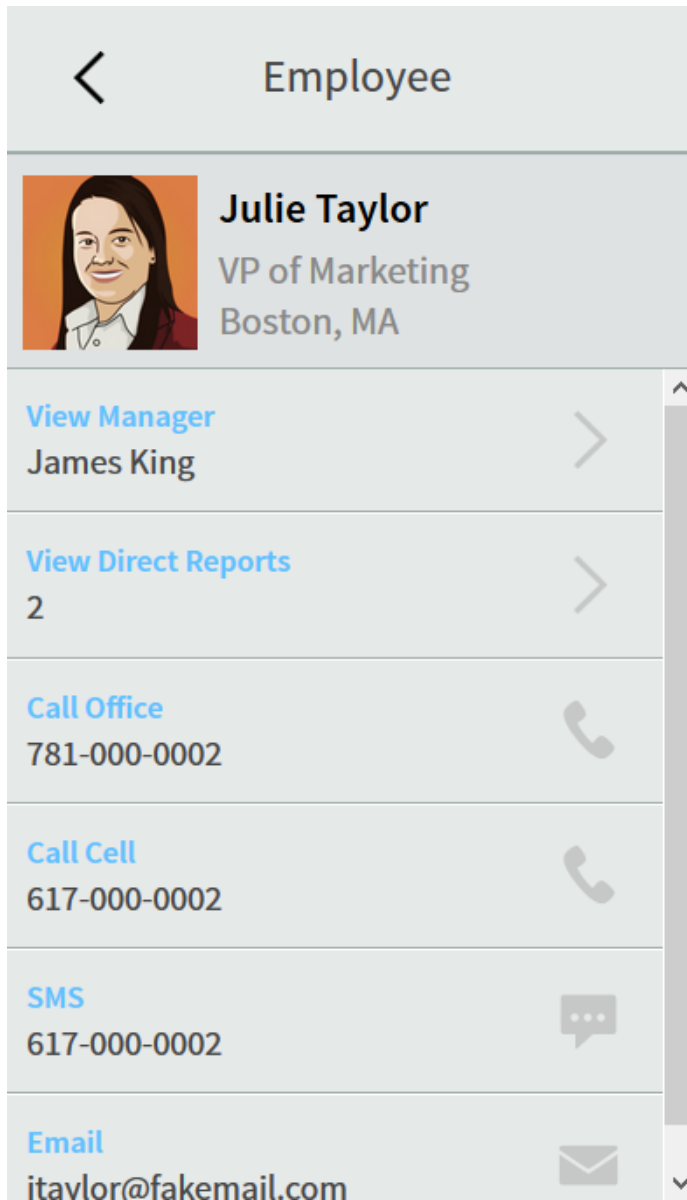
1. Home Page - List of all employees, something like:



- A list with name, picture, title, number of direct reports, and a few others (like the original user management)

- Other things: sorting, search, endless/infinite scrolling
- When an employee is clicked, it goes to his/her summary in the list.
- There should be ADD button to add a new employee.

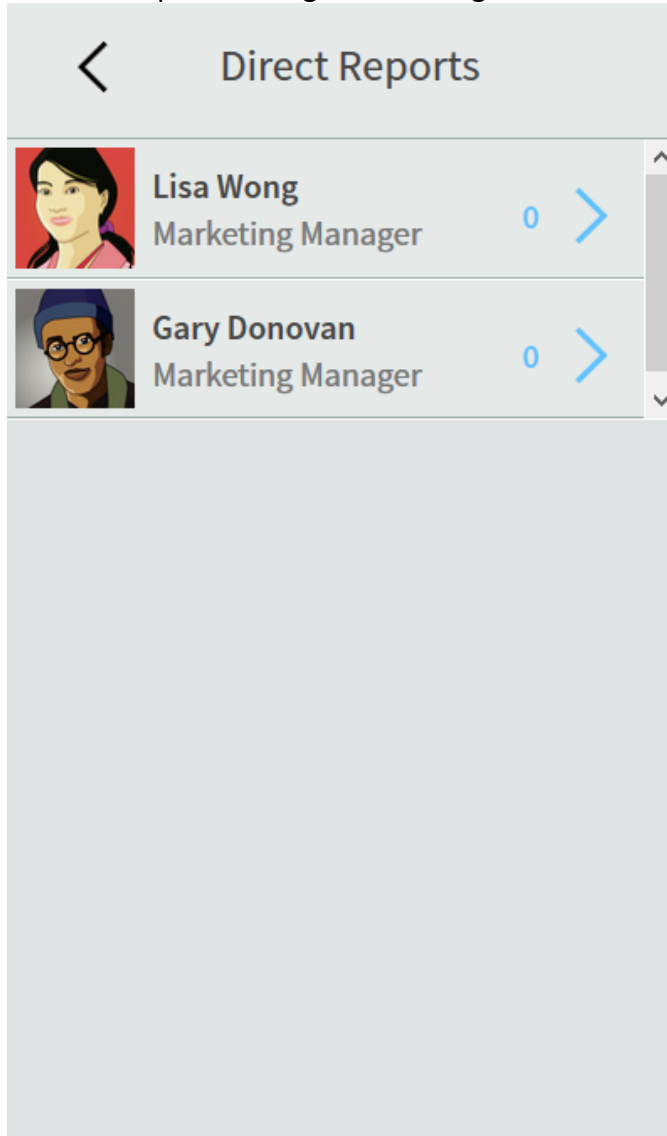
2. Employee Details: something like this:



- Show details information about the employee
- When manager is clicked, it goes to the manager's detail page,
- When Direct Reports is clicked, it goes to the direct report list page (below)
- When a contact is clicked, it makes calls or sends email

- The employee information can be edited via a EDIT button and then SAVE
- BACK button goes to the Home page

3. Direct Report List Page. something like:



- Shows direct reports name, picture, and his/her direct reports
- When a direct report is clicked, it goes to that person's detail page
- When BACK button is clicked, it goes to the original employee's detail page

For data structure, the employee object (in JSON) should have a property for the employee's manager (which is a single person) and a property for his/her direct reports (which should be an array). The backend should be implemented with nodeJS and mySQL/MongoDB that provides REST API for the employees. In this case, you can try to allow picture file upload into a static folder.

Here is a list of possible fields:

Employee

Name:

Title:

Sex:

Start Date:

Office Phone:

Cell Phone:

SMS:

Email:

Manager:

Number of Direct Reports:

Key New Features to Implement

1. please write a design document first
 - database design and reasons
 - REST API specification
 - design of each API function at NodeJS/ExpressJS
 - UI design
 - AngularJS view template, controller, service specification. Including UI mock-up if you have them
2. design and implement manager - report_to relationship, which is 1-to-M. An employee is allowed to not have a manager also. But the relationships can't form a 'circle' among employees
3. CRUD operations, especially to include the above relationship
4. photo upload (in New), change/remove (in Edit), and display (full image in Edit and thumbnail in list)
5. invoke phone and email app to dial the number / send email
6. endless/infinite scrolling on the Employee list, no pagination
7. check and prevent loops in the report_to relationship