**Organization Charts**

**Problem Statement:**

Given an organization, which has predefined designation (Director, Manager, Lead, Developer, Devops, QA and Intern) we have to provide the charts for a given employee. The charts should include employee information, his/her manager information, his/her all colleague information (excluding him/her) and then all employee reporting to him/her.

The response should be sorted with levels across different level. If the level is same, then then response should be sorted alphabetically with employee name.

Below tables has the level id and corresponding designation.

|  |  |
| --- | --- |
| Level Id | Designation |
| 1 | Director |
| 2 | Manager |
| 3 | Lead |
| 4 | Developer, DevOps, QA |
| 5 | Intern |

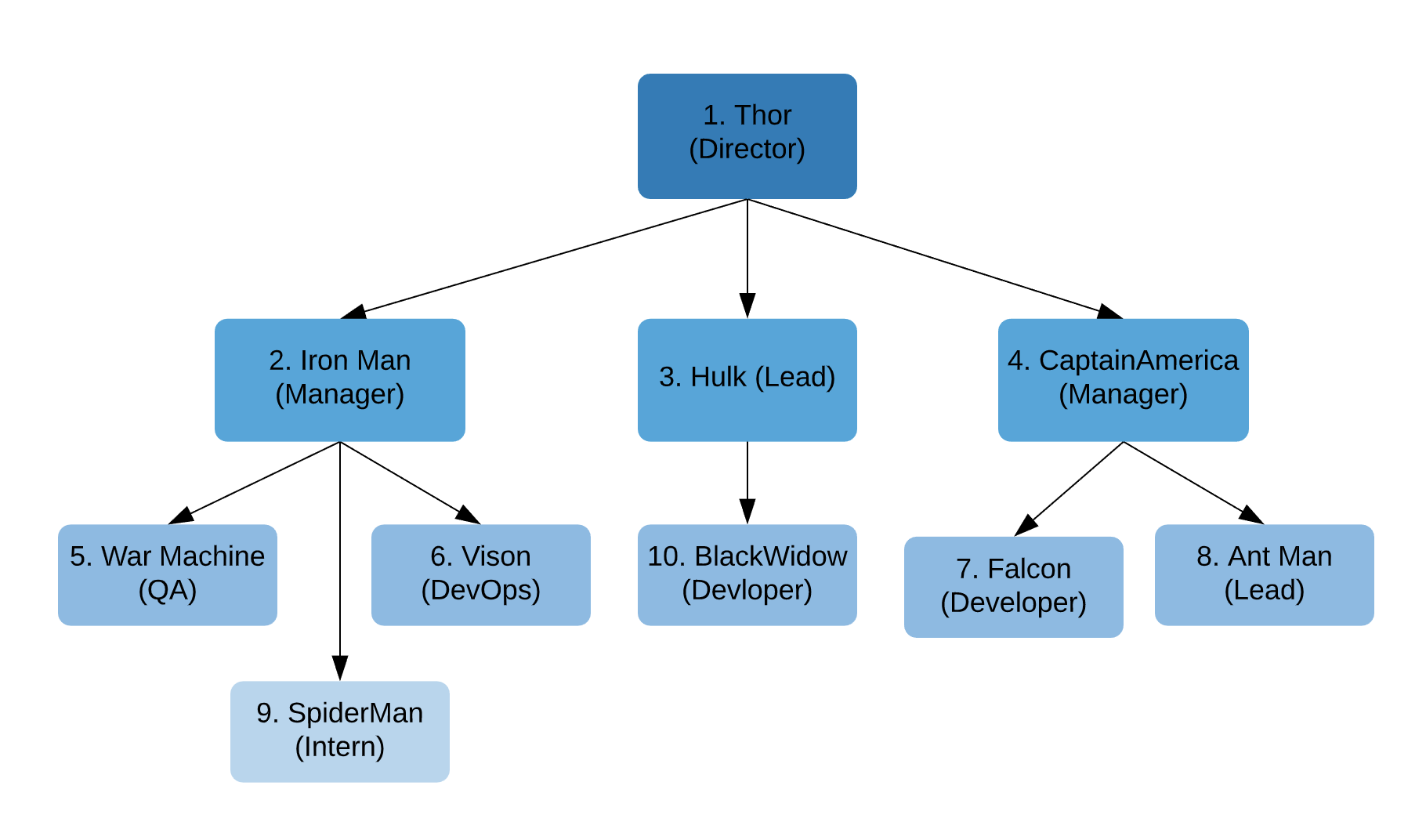
**Rest API:**

1. GET (all): Get all employee in the organization.
2. GET: Organization chart for an Employee.
3. POST: Add an employee to the organization.
4. PUT: Update an employee or replace an employee with some other employee.
5. DELETE: Remove an employee from the organization.

**Condition:**

1. Every employee will be identified with a unique id, called as **EmployeeId**. It should be generated by application. It can be kept as incremental.
2. A person can’t report to a person with designation either as same or lower in order.
3. Intern can’t have any employee reporting to him.
4. If an Employee is removed, all person reporting to him will now directory report to his manager.
5. If an Employee is moved across the organization, then all employee reporting to him will also move along with him.
6. Director can be only one across the organization.
7. Director can’t be removed (Director can only be removed if employee reporting to him is zero. That means If a director needs to be replaced, all employee reporting to him must be moved first under a different director).

**Example:**



|  |  |
| --- | --- |
| GET | For all Employee no info is required |
| For an employee employeeId will be passed |
| POST | {  "name": "IronMan",  "jobTitle": "Manager",  "managerId": 1  } |
| PUT | {  "name": "IronMan",  "jobTitle": "Manager",  "managerId": 1,  "replace": "true/false"  } |
| DELETE | employeeId |

Given employeeId as 2 (Employee Name: IronMan) below is the output. 

**Things to Focus/marking criteria:**

1. The design should in such a way that any new designation should be added to the organization without much changes.
2. Within the application variables names, Class Names, Objects name e.t.c should be self-explanatory. Provide comments if required.
3. Please follow the best standard practice for REST api.
   1. Provide proper response code with message for every negative use case.
4. Write Unit Test for your application.
5. Daily “Git commit” should be done for your project.

**Tools:**

1. Final Goal is to have a spring boot application with Spring Data JPA.
2. For starting Use Spring boot with H2Db. H2 provide an internal db, which will help in starting phase of Development.

**Deadline:**

1. **Friday (27th Sep):** Review of DB design and Rest API.
2. **Friday (11th Oct):** Final review of project.